MIL-STD-1553

455 (1)

MULTIPLEX APPLICATIONS HANDBOOK

1 May 1980

Final Technical Report

Air Force Systems Command
Aeronautical Systems Division ENASD
Wright Patterson Air Force Base, OH 45433

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS AERONAUTICAL SYSTEMS DIVISION (AFSC)
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

REPLY TO
ATTN OF: ENASF (Mr. Harold J. Alber/53586)

SUBJECT: MIL-STD-1553 Multiplex Application Handbook

TO: Defense Logistics Agency
Defense Documentation Center
Attn: Charles E. Gould
DTIC-DDA-1 82-1248
Cameron Station
Alexandria, VA 22314

1. Attached is a copy of the Multiplex Application Handbook which also contains a copy of MIL-STD-1553B. The material in this copy is identical to the previous copies distributed by SEAFAC. Recently there was a slight change from a bound handbook to one which can be placed in an ordinary three ring binder. Hopefully this will help facilitate any future changes/updates.

2. If you have any further questions, please contact any of the following engineers in the Multiplex Group: Harold Alber, Tony Haley, Duane Thorpe, Kumar Vakkalanka, Craig Burgess, or Lt Curt Adams. The phone numbers for this group are (513) 255-3586 or (513) 255-5463.

Cheryl L. Zelasco

CHERYL L. ZELASCO
Chief, Systems Engineering Avionics
Facility Branch
Avionics Systems Division
Directorate of Avionics Engineering

1 Atch
Cy Multiplex Application
Handbook

MIL-STD-1553 MULTIPLEX APPLICATIONS HANDBOOK

CONTENTS

(Chapter contents are located at the front of each section)

FL-82 on file

A

AIR FORCE/56780/6 April 1981 — 300

PREFACE

The Multiplex Applications Handbook is the final technical report required by contract F33615-78-C-0112, U. S. Air Force Systems Command, to The Boeing Company. The contract technical monitor was Mr. Erwin C. Gangl.

The Boeing Military Airplane Development organization prepared the handbook and was assisted by SCI Systems, Inc. as subcontractor. Messrs. C. Ray Turner, Boeing, and Don H. Ellis, SCI, were program managers. All contributing authors are identified below.

It is impossible to acknowledge all sources of data, information, and insight given to the authors during the handbook's preparation. Engineers and managers of many companies supplied assistance, and the multiplex literature was freely used. Special acknowledgment is due to Capt. F. L. Pensworth and Mr. Harold Alber of ASD/ENASD for their valuable advice to Boeing and SCI.

The handbook technical effort was accomplished by the following Boeing and SCI engineers with support from Messrs. Vince Angleton, Harris, David Brickner, Sperry Flight Systems, and Jim McCuen, Hughes.

Boeing	SCI
Al Crossgrove	Don Ellis
Mack McCall	Jim Gross
Lee Smith	John McGary
Ray Turner	

Handbook production was a significant accomplishment by the various secretaries and the graphics illustrator. Many persons helped in the typing, proofreading, and graphics, but special acknowledgment is due to the following people:

Susan Hickey, secretary	Boeing
Carol Scheerer, secretary	Boeing
Laura Townsend, illustrator	Boeing
Barbara Singleton, secretary	SCI
Patrice Koslow, secretary	SCI
Joan MacDonald, secretary	SCI

CHAPTER 1

INTRODUCTION

Table of Contents

List of Figures

1.0 INTRODUCTION

The multiplexing of data and the transmission of multiplexed data over a pair of twisted-shielded wires might not appear to be a broad enough subject to warrant a handbook. Nevertheless, when the ramifications of the use of multiplexed data in military aircraft are considered, the subject is indeed broad and the impact of multiplexing is far reaching.

This handbook deals with multiplexing as described in MIL-STD-1553; the current revision is MIL-STD-1553B, 21 September 1978. It is expected that the handbook will provide a greatly needed aid to understanding MIL-STD-1553 applications in both current and future military aircraft. When the handbook was first planned, the following topics were to be covered:

a. An explanation of the standard, paragraph by paragraph

b. Some historical background on multiplexing and the development of the standard

c. System design, dedicated to the system designer or architect to help decide what to multiplex, what not to multiplex, systems to be interfaced or not interfaced to the bus, redundancy levels, and bus hierarchies

d. Hardware design for the subsystem designer who has to build hardware to meet MIL-STD-1553 interface requirements

e. Software design, presenting methods of programming the bus controller(s) to make efficient use of the bus system time-sharing aspects

f. Examples, if appropriate

The need for a handbook was recognized by the various people who participated in the latest revision of MIL-STD-1553 (e.g., members of the SAE-A2K committee on multiplexing and representatives of the Air Force, Navy, and Army). They felt that the rationale for requirements in the standard, which reflect current practice, should be organized and presented to the industry and DOD as a whole.

The handbook is organized into three major areas to satisfy requirements for the planned topics and expectations of the industry and DOD. The handbook organization is presented in figure 1-1; the three major areas are--

a. Introduction and background
b. Design
c. Reference material

Section contents are placed at the front of each major section. The following sections summarize each section of the handbook.

1.1 DESCRIPTION OF BACKGROUND SECTION

Section 2.0 "Background," attempts to relate the history of MIL-STD-1553 (also referred to subsequently as 1553) to the evolving technology of multiplexing and advancing digital technology.

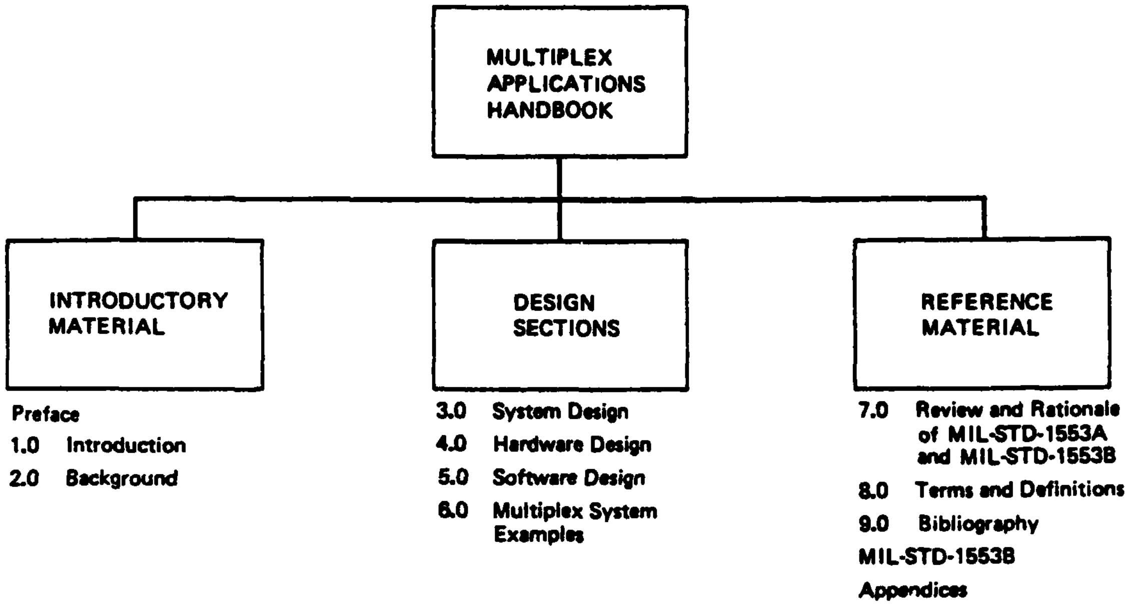


Figure 1-1. Handbook Organization

Section 2.0 includes--

2.1 The history of multiplexing and of MIL-STD-1553
2.2 Administration of standardization and the importance of 1553
2.3 Overview of the contents of the 1553 data bus standard
2.4 Technical trends in multiplexing
2.5 Issues resolved by developing the standard
2.6 Rationale for choice of data bus characteristics

Sections 2.1 through 2.4 should provide an overview for the person unfamiliar with 1553.

1.2 DESCRIPTION OF DESIGN SECTIONS

The design sections' organization is presented in figure 1-2 and includes these sections:

3.0 System Design
4.0 Hardware Design
5.0 Software Design
6.0 Multiplex System Examples

The major challenge in writing these sections was to start from the fundamentals, making the sections useful to beginners, and still include enough detail relative to system, hardware, and software design to make the material useful to experienced engineers, while limiting the subject to multiplexing. In this handbook, most of the general design advice has been alluded to or referenced but not included. For example, requirements for

DESIGN SECTIONS

3.0 SYSTEM DESIGN

- 3.1 System Topology and System Control Introduction
- 3.2 Avionics Integration Design Activities
- 3.3 Multiplex System Requirements and Design Specification
- 3.4 Multiplex System Test
- 3.5 Roadmap to Multiplex System Design

4.0 HARDWARE DESIGN

- 4.1 Multiplex Terminal Definition and Functional Partitioning
- 4.2 Bus Network/Terminal Interface Considerations
- 4.3 Bus Coupler Design
- 4.4 Transceiver Design
- 4.5 Terminal Design

5.0 SOFTWARE DESIGN

- 5.1 System Control Considerations
- 5.2 Control of Application Software in a Multiplex System
- 5.3 Multiplex Impact on Applications Software
- 5.4 Support Software
- 5.5 Bus Control Software Example
- 5.6 Processor to Bus Interface Characteristics Assumed for the Example

6.0 MULTIPLEX SYSTEM EXAMPLES

- 6.1 F-16 Multiplex System
- 6.2 LAMPS Multiplex System
- 6.3 YAH-64 Advanced Attack Helicopter Multiplex System
- 6.4 B-52 Offensive Avionics System Multiplex System
- 6.5 DAIS Multiplex System

Figure 1-2. Design Section Organization

service qualification are not directly referenced but must be considered. Standard engineering practice must also be exercised during the design process.

The initial sections in each major section introduces the subject. These paragraphs are directed to the manager or new engineer.

Section 3.0 "System Design," contains--

3.1 System Topology and System Control Introduction
3.2 Avionic Integration Design Activities
3.3 Multiplex System Requirements and Design Specification
3.4 Multiplex System Test
3.5 Roadmap to Multiplex System Design

Section 3.1 is a tutorial on architecture, which is defined as the composite of physical connectivity (topology) and system control. Section 3.2 covers the process of avionics integration as currently practiced by a typical system engineering or airframe contractor. Section 3.3 is the bridge from system design to hardware and software design, as these activities depend on allocated requirements. Test is treated briefly in section 3.4. Testing is not strictly a design activity, but it is important to understand the advantages that multiplexing brings to system test. Section 3.5 is mainly of interest to system program offices as a guide to development of a multiplex system as it relates to normal DOD system acquisition.

Section 4.0, "Hardware Design" contains--

4.1 Multiplex Terminal Definition and Functional Partitioning
4.2 Bus Network and Terminal Interface Considerations
4.3 Bus Coupler Design
4.4 Transceiver Design
4.5 Terminal Design

As in the System Design section, section 4.1 contains tutorial material. This tutorial also establishes the definitions that are used throughout the section. A multiplex terminal is a combination of analog circuits and digital circuits: therefore, the sections that follow treat various terminal design aspects. Sections 4.2 through 4.4 define the design task required to meet 1553 requirements for data transmission and reception. Section 4.5 treats all the different terminals allowed by 1553 by including descriptive material on a few basic types.

Section 5.0, "Software Design" contains--

5.1 System Control Considerations
5.2 Control of Application Software in a Multiplex System
5.3 Multiplex Impact on Application Software
5.4 Support Software
5.5 Bus Control Software Example
5.6 Processor to Bus Interface Characteristics

Section 5.1 tells the software engineers what they need to know to design software for a multiplex system. Section 5.2 is a broad narrative of system and software sequences and both normal and abnormal operation of a multiplex

system. Section 5.3 briefly discusses the allocation of software to avionic processors, the need for error-tolerant software, and provision for communication growth. Unique support software is described briefly in section 5.4. Sections 5.5 and 5.6 present a detailed example showing how software is used to communicate with a particular type of bus control hardware.

Section 6.0, "Multiplex System Examples," describes without comparison or evaluation, five current multiplex systems:

a. The F-16 Air Force fighter airplane
b. The LAMPS MK-III Navy ASW helicopter
c. The YAH-64 Army advanced attack helicopter
d. The B-52 Air Force offensive avionics system
e. The DAIS Air Force Avionics Laboratory system

1.3 REFERENCE MATERIAL SECTIONS

This area comprises eight sections:

7.0	Review and Rationale of 1553A and 1553B
8.0	Definitions
9.0	Bibliography
10.0	MIL-STD-1553B
Appendix A	Tools Needed for Data Base Analysis
Appendix B	Data Bus Use Analysis
Appendix C	Bus Network Modeling
Appendix D	Life Cycle Cost Analysis

The scope of sections 8.0, 9.0 and 10.0 is self-evident. Section 7.0 is a paragraph-by-paragraph discussion (commentary) of MIL-STD-1553B with an explanation of changes from 1553A. This commentary on 1553B should be useful to all who desire additional explanation or perspective on any part of the standard. Appendices A through D provide expanded discussions of issues briefly discussed in the System Design Chapter.

CHAPTER 2
BACKGROUND

Table of Contents

List of Figures

List of Tables

2.0 BACKGROUND

The U.S. Department of Defense (DOD) sets standards to be used and applied by the military services and their contractors. A military standard is a document that establishes engineering and technical requirements for processes, procedures, practices, and methods that have been adopted as standard. One of these, MIL-STD-1553, "Aircraft Internal Time Division Command/Response Multiplex Data Bus," has been in use since 1973 and is widely applied. MIL-STD-1553 will be referred to as "1553" with the appropriate revision letter (A or B) suffixed to 1553. The purpose of this section is to describe the development, use, and refinement of 1553 data bus standardization. Brief comments on each word in the title of 1553 are as follows:

a. Aircraft. This word denotes that the data buses are installed in DOD aircraft. Although there are instances of 1553 data buses used in ground applications, the characteristics of the bus have been determined by aircraft requirements.

b. Internal. This word denotes that the data bus is used for transmission only within an aircraft.

c. Time-Division Multiplex. The type of multiplexing for which the standard establishes requirements. Time-division multiplexing is the transmission of information from several signal sources through one communication system with different signal samples staggered in time to form a composite pulse train. A transmission line known as a twisted-shielded wire pair is used for transmission; consequently, all messages must be transmitted and received serially. The 1553 data bus is sometimes referred to as the 1553 serial data bus.

d. Command/Response. This denotes the communication protocol that distinguishes 1553 data bus operation from other data buses. Basically, the command/response protocol described in 1553 is a "speak only when spoken to" communication that is solely the responsibility of a designated controller.

To cover the background of 1553, the subsections of this chapter discuss the following related topics:

2.1 The history of multiplexing and MIL-STD-1553
2.2 Administration of standardization and the importance of 1553
2.3 Overview of the contents of the 1553 data bus standard
2.4 Technical trends in multiplexing
2.5 Issues resolved by developing the standard
2.6 Rationale for choice of data bus characteristics

Section 2.1 describes the influences that advancing technology and new airplane programs and their avionic systems had on the use of multiplexing. It also gives a detailed chronology of 1553 evolution. In section 2.2, the Air Force's general plan for administration of avionics standardization is presented, and the transition to 1553B is discussed. Section 2.3 is an overview of 1553B. The major requirements are summarized, and the interrelationship of the sections is presented. This section should help new readers of 1553 understand the overall importance of the requirements.

Section 2.4 is a loose aggregation of subtopics that presents reasons for the use of 1553. Section 2.5 briefly points out the ease and broad application of 1553 and that de facto standard terminal types are available. The purpose of the section is to demonstrate that 1553 multiplexing has earned its wings and is ready for a wide range of use. Finally, the rationale for the choice of electrical characteristics is presented, as it is really the "technical history" of why the unique electrical and coding techniques of 1553 were selected. This chapter begins with the top-level view and ends by helping the curious electrical engineer understand why Manchester biphase, 20 bit times for a 16-bit word, and transmission over a "lossless" line are good.

2.1 HISTORY

The history of multiplexing in aircraft will be discussed from two viewpoints: (1) the requirement to multiplex because of air vehicle avionics integration complexity and (2) the standards development that was the forerunner to MIL-STD-1553B and this handbook.

2.1.1 Air Vehicle Avionics Integration

Avionics integration, which is defined here as the cooperative use of shared information among avionic subsystems, first became a necessity when requirements for missions and their associated avionic hardware could no longer be met practically in air vehicles with independent and self-sufficient subsystems. Elimination of unnecessary duplication of information sensing and display, performance gains, reliability gains, cost reduction, and lack of space are usually given as the major reasons for integration. Subsystems were forced to depend on each other for basic information. This level of integration began with the most complex subsystem because it had the most capability, as well as the most need for information from other subsystems. As digital technology progressed, this central subsystem was expanded to incorporate mission processing (processing not specifically associated with a subsystem or display). However, problems arose early in the centralization approach because subsystems were designed with no concern for interconnection with other subsystems. Each subsystem had been specialized, and the interfaces reflected this specialization. The central computer input-output (I/O) circuitry was designed to perform the functions of ordering this incoming and outgoing data, and the computer was often small compared with the size and complexity of the I/O. Even so, the central computer concept and its associated integration upgraded the capability of the mission and made sensible use of shared information. It was then reasoned that some of the centralization problems related to the complexity of the I/O could be solved if the I/O circuitry could be partitioned and distributed, alleviating the central unit's complexity. Commercial data transfer trends supported this view. Multiplexing, which makes information transfer convenient and simplifies I/O, offered this capability, and the extended computer I/O philosophy was developed. Multiplexing makes information exchange convenient because sensors and processors are all "on the bus." Multiplexing simplifies I/O because the information transfer medium is reduced to a single wire pair. This extended I/O philosophy was adopted extensively by military avionics integrators with the development and use of military minicomputers and the availability of lower cost digital components. These avionics integration methods began to be referred to as multicomputer systems. This made possible the

distribution of the computation and permitted several computers to replace the more powerful central processor. In addition, these multiple computers added desirable redundancy features; MIL-STD-1553(USAF) and the Digital Avionic Information System (DAIS) were developed using this integration concept. Application of this concept in various forms exists today on several aircraft (e.g., B-1, F-16, F-18, and Space Shuttle). From the subsystem equipment point of view, these approaches to integration use both integration units for unmodified subsystem interfaces (interface boxes that 1553 calls remote terminals) and embedded interfaces (1553 interface circuitry housed within a subsystem box). These remote terminals (RT) or embedded 1553 interfaces connect the subsystems to the data bus(es). The current trend of embedding the interface in the subsystem is expected to continue.

The integration approach using multiplexing is implemented by defining information transfer formats and electrical interface characteristics. Therefore, the functional performance is accomplished by both hardware and software. Most of the problems associated with the centralized I/O have been eliminated by this approach, while others have surfaced (e.g., software complexity, synchronous operation, multiple executive control, data communication, and I/O circuitry). But with all this, a decided improvement over previous approaches has been achieved. Technology improvements in computers and digital hardware (i.e., microprocessors) and maturation of the software design process allow further extension of the integration approach by a more distributed system concept consisting of both microcomputers and minicomputers. The newer integration approaches will use more processors and buses to functionally partition the avionics along common military and industry organizational lines (such as navigation, stores management, control and displays, and communication). This functional partitioning should further ease the integration problem by allowing design of the functions to be developed more independently of each other prior to completing the total avionics integration.

2.1.2 MIL-STD-1553 Chronology

Development of a standard digital time-division multiplex data bus began in early 1968 and continued through 1978 with the latest revision (MIL-STD-1553B). The Society of Automotive Engineers (SAE), Aerospace Branch, established a subcommittee of industry and military personnel in 1968 to define some of the basic requirements of a serial data bus. By this means, an exchange of industry and military views was accomplished. The committee, Multiplexing for Aircraft (SAE-A2K), developed the first draft of a data bus standard that was similar to the present military standard. It represented a mixture of military standard requirements and procurement specification requirements. Its format allowed standardization on requirements that could be agreed upon and a slash sheet in the appendix for requirements that appeared to be vehicle particular. This document represented the best that the industry and the military could define at the time. The benefit of this document was that it produced a sounding board for ideas. In this respect, it was successful and provided the step forward required to develop the USAF military standard, MIL-STD-1553, in August 1973 and the USN/Specification, "Control Group, Electric Power, OK-XXX-(V)/A, General Specification for," MIL-P-81883, in May 1976. During the years from inception of the SAE-A2K to the release of the first military documents, the industry was designing and producing hardware for various multiplex

systems. Some of these systems were developed prior to or during the standardization era (e.g., F-15 and B-1). Because of program timing, each system went its own way because no standardization effort existed at the time. However, with the production of the F-16, MIL-STD-1553(USAF) found its first full aircraft application. From 1973 to 1975 (when MIL-STD-1553A was released), industry and the military (Air Force, Army, and Navy) coordinated their efforts to determine the degree of standardization required. During this time, several preliminary drafts of Air Force and Navy documents were developed and extensive industry comments were solicited.

By late 1974 and early 1975, the DOD directed the military to develop a single position and to make the necessary revisions to MIL-STD-1553(USAF). Based on this effort, 1553A was released in April 1975. Since 1975, industry and the military have continued to coordinate the standard through symposia, studies, and military development programs. With the standard available, the industry and the military began to apply the data bus to more operational vehicles and systems. As applications became extensive, certain difficulties were recognized in MIL-STD-1553A.

Discussions concerning these difficulties were conducted between the SAE-A2K and the DOD Triservices Committee (the group responsible for controlling the military standard). These discussions resulted in the formation of an SAE task group (MIL-STD-1553 Update) in October 1976. The task group's assignment was to develop suggested changes to 1553A. Once again, a task group was formed from several industry and military segments. The task group solicited comments from industry and the military to support its work. These responses were extensive and involved foreign as well as domestic equipment suppliers and users of the standard. It was from this base that the task group developed and presented the suggested revisions to 1553A. In October 1977, after review and discussion of suggested changes, the SAE-A2K approved a proposed revision; in December 1977 these recommendations were provided to the DOD Triservices Committee. In addition to the SAE input, industry comments on changes to 1553A were solicited in January 1978 by the DOD Triservices Committee. Based on these comments, the DOD Triservices Committee met on several occasions and produced a draft of 1553B. This draft was presented to the SAE's task group in April 1978 for review and comment. Following this review, one final meeting was held with task group members in June 1978, during which final agreement between the SAE task group and the Triservices Committee was obtained. From these verbal agreements, final written approval was sought within the Triservices Committee and, upon receipt of the written approvals, MIL-STD-1553B was released as an official document on September 21, 1978.

2.2 ADMINISTRATION OF STANDARDIZATION

The problems that the DOD is facing today in avionics exist in three basic areas: excessive cost, low reliability, and lack of standardization.

The high cost associated with the development, procurement, and support of new weapon systems and the associated flow time required to accomplish an avionics system evolution are limitations to the approval of new needed systems. In addition, the problems of poor field reliability and the difficulty of repairing existing avionics that were not designed to a common standard have decreased the effective level of the operational units. This lack of standardization has fostered an environment in which each new weapon system can require all new hardware, software, and support equipment.

Each problem has been recognized by the Air Force, and avionics strategy, policy, and plans are being implemented to reverse the trends. The basic USAF strategy is to establish goals for reducing avionics proliferation and cost while increasing avionics availability. To achieve these goals, the Air Force has prepared and adopted Air Force Regulation 800-28, "Air Force Policy on Avionics Acquisition and Support," which establishes a charter and a mode of operation for the Deputy for Avionics Control. This single USAF organization will be responsible for focusing and controlling all USAF avionics efforts. It will be staffed jointly by the Air Force Systems Command and Air Force Logistics Command to --

a. Develop and maintain a USAF avionics master plan and an associated data base containing cost and technical data on all avionics inventory items

b. Guide all Air Force Systems Command and Air Force Logistics Command avionics activities by reviewing and approving all avionics plans

c. Assist development of a USAF avionics investment strategy to reduce procurement cost of avionics

d. Review and coordinate all avionics programs to demonstrate compliance with the regulation

Several specific objectives are being implemented. The first objective involves promotion of standards to provide avionics architectural interface commonality across systems. This is being accomplished by the use of MIL-STD-1553B, "Aircraft Internal Time Division Command/Response Multiplex Data Bus." The second objective is to promote commonality of Air Force Integrated Support Facilities (AFISF). This is being accomplished by the use of higher order language (HOL) software and the imposition of architectural interface standards in all major weapon systems. Commonality of test support is expected.

The third and fourth objectives in this process are to develop standardized group B avionic equipment and subsystems and their associated common support (test) equipment. This category of group B avionic equipment and subsystems have been defined to include inertial navigation, computers, pod-mounted sensors (e.g., FLIR, LLTV, and PLSS), weapons guidance, radar, multiplex data bus, control and displays, man-machine interfaces, flight control, radio navigation aids, electronic countermeasures, communications, antenna, and avionic test equipment. The fifth objective will be to recommend priorities for research and development programs to ensure that these objectives are met. The sixth objective is to increase availability of avionics by investing in reliability and maintainability programs. The seventh and last objective is to examine this standardization effort from an overall USAF view to ensure that desired interoperability is achieved across weapon systems. Based on this strategy, the USAF regulation, and the plans that are being implemented, the Air Force intends to achieve the needed standardization across weapon systems, thus reducing their cost and proliferation and allowing improved reliability by use of common elements over a longer period of time.

2.3 OVERVIEW OF THE CONTENTS OF THE 1553 DATA BUS STANDARD

The purpose of this section is to present enough of the organization, scope, and contents of the 1553 standard to assist readers unfamiliar with what the

standard covers. The current version of 1553 (namely 1553B) will be discussed. Key terminology of 1553 will be introduced and explained.

2.3.1 Scope of the Standard

The standard establishes requirements for information transfer formats and electrical interface characteristics. In section 2.1.1, "Air Vehicle Avionics Integration," the mechanization of an integration using 1553 was described as accomplished by both hardware and software. The information transfer formats are originated and interpreted with software. The electrical interface characteristics describe the data transmission technique. These electrical requirements can be isolated from the remainder of the content of the standard. In fact, the SAE-A2K Fiber Optics Task Group did analyze 1553B to determine what paragraphs could be common to a standard for both wire transmission (as defined in 1553B) and fiber-optic transmission. Those portions of 1553B that define electrical characteristics are often described as the "specification portion" of the standard. It is proper to view 1553B as part standard, part specification. To obtain a standard interface that could produce commonality among manufacturers, the descriptions in those portions of the standard are similar to a military specification. Therefore, the electrical characteristics and the information transfer formats will be discussed separately in this overview.

Before proceeding to discussions of these two essential sets of requirements in 1553B, some of the ancillary topics and their relationship to information transfer formats and electrical characteristics need to be covered briefly.

2.3.2 Summary of the Contents 1553B Section

The organization of the standard is --

Foreword
1. Scope
2. Referenced Documents
3. Definitions
4. General Requirements
5. Detail Requirements
10. Appendix

This organization complies with the format established for military standards. Several points need to be made with respect to the interrelationship of the sections:

a. The "Foreword" establishes an application note that is very important; namely, that 1553 is a set of "...techniques which will be utilized in systems integration of aircraft systems." In effect, 1553 is synonymous with a method of integration as well as an interface standard.

b. The "Definitions" section is itself a standalone tutorial, in which the order of the definitions was carefully considered. The idea was to introduce concepts in a progression that allows building of new usages from previous definitions.

c. The "Scope" and "Referenced Documents" sections are necessary but not substantive.

d. All 1553B requirements are general; there is no text in section 5, "Detail Requirements."

e. The need for explanation and insight into the standard was recognized, and section 10, "Appendix," is a very brief attempt at documenting some advice and systems considerations on these six subjects: redundancy, bus controller, multiplex selection criteria, high reliability requirements, stubbing, and use of broadcast option. This handbook provides an extensive supplement to that section.

2.3.3 Information Transfer Formats

The term or phrase "information transfer formats" is used in 1553B interchangeably with "message formats." The exchange of messages in 1553B is very precisely described, and there are only 10 allowable formats. If an exchange cannot be completed because of hardware or software failures, then 1553 describes and specifies what is to be done. All methods of followup to retry the message or to determine the failure must be done within the allowable 10 message formats. It is this idea of proper exchange of messages that makes it appropriate to refer to them as "protocol" -- because it is similar to the process that diplomats use to exchange state notes. Frequently the message format requirements of 1553 are also referred to as 1553 protocol. Message formats are composed of words, response time gaps, and intermessage time gaps. There are only three types of words: command word, status word, and data word.

Message formats are divided into two groups: those that are essentially reserved "...to communicate with the multiplex bus related hardware, and to assist in the management of information flow..." and those that are essentially used to "...extract data from and feed data to a functional subsystem." The use of some of these features is optional; that is, if chosen in the application of 1553, there is a minimum prescribed meaning to each of the bits in some words, that must be used if those bits are designated by the designer to be used.

Again, the two groups of message formats are mode commands and data transfers.

2.3.3.1 Mode Commands

Mode commands are those formats reserved for communication with the bus hardware and information flow management.

There is provision for 32 unique mode commands, and 1553B specifies the base 2 numbers that are to be used for 15 of these. The balance are reserved, which means the designer must secure special approval to use a reserved mode command number. However, the use of any or all defined mode commands is optional.

2.3.3.2 Data Transfers

Data transfer message formats, on the other hand, do not restrict the designer to the same degree as mode commands. The restrictions are (1) no

more than 32 words in any single message are to be used and (2) the most significant bits of any value or quantity will be transmitted first, with bits of descending significance following.

2.3.4 Modes of 1553 Terminals

Before concluding the discussion on information transfer formats, it is necessary to describe the modes of terminals allowed by 1553. A terminal in 1553 parlance is "the electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus... ." There are only three functional modes of terminals: the bus controller, the bus monitor, and the remote terminal. The definition of a terminal as an electronic module should convey the notion of a unit that contains digital logic as a minimum and may frequently contain microprogrammed LSI or a microprocessor with instructions in ROM. As a bus monitor or bus controller, the usual approach is a connection to and a dependence on a minicomputer for functional performance. Significant digital complexity is required because 1553 specifies response time and data storage requirements that require dedicated digital hardware.

2.3.4.1 Bus Controller

The "Definitions" section states that the bus controller is "the terminal assigned the task of initiating information transfers on the data bus." Other requirements are: (1) "The bus controller is the key part of the data bus system," and (2) "Sole control of information transmission on the bus shall reside with the bus controller... ." These quotes clearly define the bus controller mode.

2.3.4.2 Bus Monitor

The standard defines the bus monitor as "the terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time." Bus monitors are frequently used for instrumentation.

2.3.4.3 Remote Terminal

Any terminal that is not operating in either bus controller or bus monitor mode is operating in the remote terminal mode.

2.3.5 Types of 1553 Words

The 1553 standard allows only three types of words, and the term "format" is also used to describe the allowable meaning of the bits. In 1553, "...a word is a sequence of 16 bits plus sync and parity." Each word contains the sync, which is 3 bit times (but not 3 bits), and parity, which is 1 bit, so that transmitted words in a 1553 system are always 20 bit times in length for each 16-bit word. The role of each of the three allowable word formats is as follows:

a. Command Word. This word is always used as the first word (or words) of a message. Therefore, it will only be transmitted by a bus controller. This word defines the type of information transfer format that will be used.

b. Status Word. This word is always used as the first word that is transmitted by a remote teminal. (Bus monitors do not transmit at all.) This word contains the status of the transmitting remote terminal.

c. Data Word. This word (or words) is always transmitted contiguously with a command word, status word, and other data words.

2.3.6 Electrical Characteristics

The key characteristics of the 1553 data bus are as follows:

a. Data Rate. "The transmission bit rate on the bus shall be 1.0 megabit per second... ." Accuracy, short-term stability, and long-term stability are specified.

b. Data Code. "The data code shall be Manchester II biphase level." This is the form of the logic 0's and logic 1's.

c. Serial Transmission of Data. "The signal shall be transferred over the data bus in serial digital pulse code modulation form." Simultaneous transmission and reception by a bus controller and a remote terminal are not possible on the same data bus. If the serial transmission of data causes an unacceptable data lag, delayed response, or capacity problem, additional buses may be used.

d. Sync. "The sync waveform shall be ...three bit times, with the sync waveform being positive for the first one and one-half bit times, and then negative for the following one and one-half bit times." This definition is for the command word and status word syncs. Data syncs are initially a negative pulse followed by a positive puls^. There is no separate clock line or source for synchronizing the receiver's terminal.

e. Intermessage Gap. "The bus controller shall provide a minimum gap time of 4.0 microseconds between messages... ."

f. Response Time. "The remote terminal shall respond ...to a valid command word within the time period of 4.0 to 12.0 microseconds."

g. Hardware Characteristics. This section of 1553B defines the following terms in specification language: cable and cable impedance, attenuation, termination, cable stubs (direct and transformer coupled), and terminal input and output characteristics (waveform, noise, symmetry, common mode rejection, and impedance).

2.4 TECHNICAL TRENDS IN MULTIPLEXING

The use of integration to improve the vehicle's mission performance has caused the vehicle avionics integrator to look at multiplexing as a method for achieving integration. In other words, the system design has become indistinguishable from the system design method. Also, there is great emphasis in the present military environment to extend the airframe life, thus causing the avionics system integrator to examine new approaches to avionic systems that will provide design flexibility to meet evolving missions and future threats. Integration using data buses is an important

step in reaching these goals. This section will identify these issues and the approaches developed.

2.4.1 Vehicle Mission Capability Improvement

Improvements in mission capability provided by integration can be classified into four categories:

a. Performance gain by the effective use of dissimilar or similar subsystems to reduce the effect of systematic error sources within a given subsystem; for example, Kalman digital filtering of Doppler and inertial navigation subsystem data can be used to obtain smoothing and prediction.

b. Reduction of total hardware by using a sensor to provide common input data rather than dedicated input sensors for each subsystem or display.

c. Effective redundancy without massive hardware duplication made possible by the integration of identical, similar, or dissimilar sensors to make multiple sources of similar data available; this is called dissimilar sensor redundancy.

d. Weight reduction and flexibility of integration and test are achieved by using multiplexing.

The first three categories are common to an integration, whereas the fourth category is associated with the method or approach taken to achieve integration. The advantage of using the serial, time-division multiplexed data bus approach to integration will be briefly discussed.

Weight saving is achieved by the reduction of wire weight provided by the serial multiplexing of digital data as compared with the point-to-point unidirectional interconnection required to achieve similar integration without the data bus. The data bus provides a path upon which many users can communicate with each other without requiring a dedicated link to each other. Weight savings vary greatly among the systems being compared with the data bus. If an analog system with analog point-to-point wiring is compared with a digital multiplex system (1553), considerable wire weight savings can be achieved. This weight saving will be reduced somewhat if the analog sensors and displays are interconnected with integration units that interface these sensors and displays with the data bus. In other words, the overall weight saving resulting from the reduction of aircraft wiring is offset by the weight of integration units. However, if the subsystem is digital and compatible with the bus interface, the offset is recovered. Another comparison of weight saving (but not as great as in the previous case) is a digital system that uses digital point-to-point data interconnections versus 1553 data bus use. When the digital system is compared with a 1553 approach to integration, the advantage is in the multiple access provided by the data bus in contrast with the point-to-point interconnects previously required. Therefore, smaller gains are achieved because both systems use integration and multiplexing in slightly different ways. Each example represents extremes in weight savings. Most new and existing systems will exist within these bounds with a mixture of both types, thus providing varying weight savings dependent on the actual use.

The integration flexibility that is available in 1553 systems is one of the key features of this method of integration. Because of the common serial interface, the high data rate (up to 50,000 words per second), the multiple access, and the command/response data format, the 1553 integration provides extensive flexibility in the development period as well as during the operational time period.

Other digital integration methods have failed to meet the flexibility requirements necessary in the military environment. These failures occurred because of the following reasons:

a. Too low a data rate was selected (data rate selected based on initial need with little growth capability)

b. Insufficient definition of interface (difficulty in duplicating the interface)

c. No method for expansion to new sources or deletion of sources (inflexible to hardware additions or deletions)

d. Limited data encoding and decoding capability (e.g., restricted to BCD or ASCII)

e. Limited addressing capability

f. Inefficient data transfer (too many wires, too much overhead per data word)

g. Difficult to simulate, which would provide confidence prior to hardware development

Each deficiency was carefully considered during the development of 1553. The detailed electrical interface definition of 1553B provides the necessary requirements information to allow multiple suppliers to build compatible interfaces. The multiple access and high data rate allow extensive integration of complex systems. The capability to simulate any part of an integration using a system integration laboratory prior to hardware and system design commitment reduces the risk of new developments and modifications. The ability to communicate data in a "transparent" fashion (i.e., the 1553 system manages the communication transfer without affecting the data) is an advantage to the user. Thus the data user can encode data to the user's required format and not to the transfer system's format. The use of message addressing per 1553 rather than word addressing allows much more flexibility than can be achieved with the word addressing formats used in some point-to-point digital communication approaches. A final advantage of this approach to information transfer is the ability to control data flow in a scheduled manner from one location; namely, the bus controller. Changes in the integration can be handled by message changes in the bus controller rather than by wiring and hardware changes to the subsystems. Also, the benefit of a synchronous schedule of data transfers can ensure data arrival when it is required and not on an asynchronous, uncontrolled basis.

2.4.2 Vehicle Modifications

The concept of multimission roles for a single airframe (or a restricted family of airframes) has become a major element in our military weapon planning. Threats, which are changing more rapidly than ever before, make it necessary to plan for mission-adaptive and threat-adaptive avionics systems over the life of an airframe. Two multimission concepts are emerging. One approach is to design a core set of avionics and peripheral avionics so the avionics suite can be readily changed by removing and replacing mission-dependent functions (peripheral avionics). Another approach is to depend on established interface standards (e.g., standard hardware and software modules) that permit an avionics system to be updated (retrofitted) throughout the life of the airframe. Obviously, these approaches are not mutually exclusive, but they can be complementary.

In the past few years, it has been the goal of the DOD to develop and apply methods and technologies that would permit avionic systems to be developed with the attributes of being easily constructed, modified, and operationally verified as mission needs change. It is anticipated that such systems would lower life cycle cost and would have the flexibility required to meet changing mission requirements. A lack of interface commonality between avionic systems has made the task of system design and integration, as well as the task of upgrading or modifying systems, very costly. MIL-STD-1553 was established by the Air Force to investigate and develop standard interfaces between the various elements of the avionic system to simplify integration and reduce cost of proliferation. These concepts were demonstrated by the DAIS and are now mature. They are being applied on several near-term programs and are planned for all future systems.

2.4.3 Digital Technology as it Applies to Integration

The technology revolution of the 1970's has provided the only method to meet the avionic system and vehicle integration requirements previously described. These changes have been accomplished because of the development process occurring in the semiconductor industry. Development and use of large-scale integration (LSI), particularly in the area of programmable devices (microprocessors), have been the keys to obtaining digital subsystems, integration devices, and displays. In addition to LSI development, the extensive use of hybrid circuitry to bridge the analog-to-digital interface has brought the avionic system integrator a set of digitally based hardware. These two factors alone have allowed the development and implementation of integrated avionic systems.

Examination of the future impacts of this technology indicates that the general trend is toward fewer, more flexible devices. As the number of unique devices declines, because of the expanded use of LSI, the requirement for flexibility can only be met by programmable devices. To maintain this decline in unique components, the semiconductor industry will look toward the user industry for widely used, good standards to define these devices. Based on the stability and producibility of these standards, the industry will make decisions to produce certain devices. Therefore, it is essential that these standards are as technology independent as possible to prevent obsolescence within a few years, since these standards will set the stage for the semiconductor industry to produce devices that will be applied to new avionic systems by avionic subsystem manufacturers. Because the

semiconductor industry must depend on standards to produce products acceptable to the military, military involvement is essential during this process of standards generation and maintenance. This has been recognized by the military, and several standards have been developed.

2.4.4 Signal Range and Type

Any data and control communication required by a subsystem can use the 1553 data bus system for communication. The only requirement is that the subsystem have at least one bus interface unit to interface with an existing 1553 multiplex data bus. Terminals that interface subsystems may have multiple bus interfaces and therefore possibly communicate on more than one bus for the purpose of redundancy, isolation of functions, or increasing the data transfer capabilities of the system.

There are classes of signals that are not generally considered for use in multiplex data bus communication. These signal classes are as follows:

a. High signal bandwidth data (single signals above 400 Hz), such as unsampled audio and video.

b. Signals used to control startup of the system prior to initialization of the multiplex system.

c. Backup signals that are required for safety of flight in the event of complete multiplex system failure. These include emergency jettison of weapons, backup flight instruments, backup communications control, backup propulsion, and backup flight control.

2.5 ISSUES RESOLVED BY DEVELOPING THE STANDARD

2.5.1 Application Areas

The intended application of the data bus standard includes data communication techniques that require (1) a command/response format, (2) a time-division multiplexed data transmission technique, and (3) application internal to an air vehicle. This has been accomplished with the application of the standard to system designs that accomplish (1) integration of air vehicle functional groups such as navigation, weapon delivery, flight control, propulsion, stores management, defensive systems, communications, and control and displays and (2) integration of these functional groups into a weapons system. The application of these system designs to various vehicles includes fighters, bombers, helicopters, and transport aircraft with missions of attack, transport, reconnaissance, and defense. It has therefore been demonstrated that the 1553 approach to integration has been proved applicable to a wide range of air vehicles, avionic functions, and missions.

2.5.2 Multiplex System Terminals

The system topology (physical arrangement) and system control will vary, depending on the intended application, the system design requirements, and the economics of the implementation; but a mix of three basic types of terminals will usually be found in any 1553 system. Terminals are built in a variety of designs and with varying capabilities. Functionally, terminals

may be bus controllers, bus monitors, or RTs. Figure 2-1 presents three types of terminals with differing levels of capability. The first terminal type is the standalone processor/bus interface unit (P/BIU). This type has the highest capability. It has the capability of being a primary or backup bus controller, bus monitor, or RT. It has full error-handling and recovery capability. The second terminal type is the smart RT. This type has less capability than the standalone P/BIU. It has no bus control capacity, unless there is sufficient processor capability to support the bus control function. Self-test is its primary error-handling and recovery mechanism. Some of the subsystems connected to this type of terminal may have processors. The third type of terminal is a low-capability RT. These RTs may be embedded in the subsystem or housed in a separate unit from the subsystem. These RTs have no bus control capability. Self-test is their only means of error handling and recovery. The subsystems interfaced to these terminals may have processors. There are several hybrid examples where a bus controller and an RT are housed in the same unit.

The multiplex system may use a single multiplex bus, redundant buses, multiple active buses, or multilevel bus structures. Redundancy can be provided through active or passive physical resources (terminals or buses) or through reconfiguration of system resources after a resource failure. The number of terminals in the system is limited by the electrical interface characteristics of the transmission medium and the bus addressing capability. Adequate flexibility of system architecture with few types of interfacing approaches is achieved. MIL-STD-1553 is not a limiting standard in this respect.

2.6 RATIONALE FOR CHOICE OF DATA BUS CHARACTERISTICS

The purpose of the standard was to develop a degree of compatiblity that would allow interchangeability of common functions built by various manufacturers, integration of various functions into a data bus system, and operational capability of a function applied to various applications. To accomplish this, the following investigations and decisions regarding the data bus system were required:

a. Modulation and coding techniques
b. Signaling methods and signal detection techniques
c. Transmission media considerations

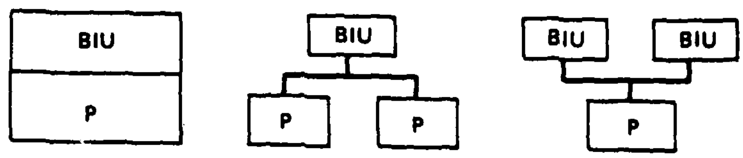
After investigation, analysis, test, and development, a set of data bus characteristics were developed or evolved to satisfy the purpose of the standard. The characteristics are described in MIL-STD-1553B. This section provides a brief review of the rationale for selecting techniques now in MIL-STD-1553. The material in this section was generally derived from "A Study of Multiplex Data Bus Techniques for the Space Shuttle," final summary report, No. 2635-M15, prepared for NASA/MSFC by SCI Systems, Inc., Huntsville, Alabama, November 1972.

2.6.1 Modulation and Coding Techniques

Two modulation techniques were available for use for a data bus system:

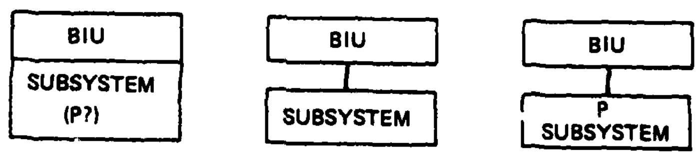
a. Baseband modulation (selected method)

Standalone Processor/Bus Interface Unit



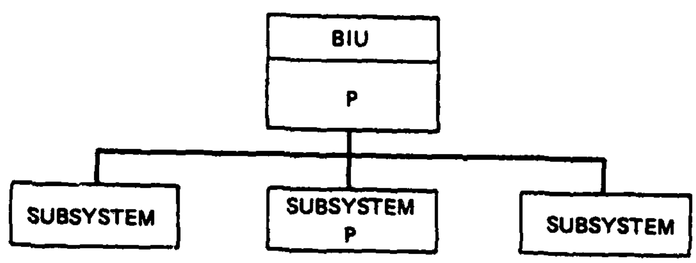
- Bus control capability (primary or backup bus controller)
- Error handling and recovery

Low-Capability Remote Terminal



- No bus control capability
- Self-test is only error handling and recovery
- Subsystem may have processor

Smart Remote Terminal



- No bus control capability unless sufficient processor capability
- Self-test prime error handling and recovery
- Subsystem may have processor

Figure 2-1. Types of Terminals

b. Carrier (amplitude shift keyed (ASK), frequency shift keyed (FSK), and phase shift keyed (PSK) modulation)

Baseband modulation is the set of modulation techniques that produce power density spectra that extend to or near 0 Hz. Carrier modulation is the set of modulation techniques whose positive spectra are symmetrical about a carrier frequency greater than zero. Baseband was selected because of its advantages over any of the carrier modulation techniques (ASK, FSK, or PSK). The difficulties associated with the carrier technique are the greater bandwidth associated with the transmission media, the hardware complexity associated with individual frequency bands for each user, the complexity of generating the carrier, and the complexity of the modulation and demodulation process.

The message format arrangement (coding) was studied to determine the most appropriate means to achieve --

a. Word or message synchronization
b. Bus access control
c. Message formatting (message types, bit patterns, error checking)
d. Channeling methods

Word Synchronization

Four methods of word synchronization are shown in figure 2-2 with their relative ranking with regard to noise immunity and bit overhead requirements. The synchronization format may be used to preface words (16 or more bits) or several word messages (32 words or greater). Also, word synchronization waveforms can be used to distinguish command data from data words. Notice that the selection of the waveform must consider both its immunity to noise and also its overhead requirements. Figure 2-2, method D, was selected as the most appropriate word synchronization method for the data bus system.

Bus Access Control

Several bus control schemes are appropriate for data bus systems. Four basic types were investigated:

a. Central control (timer referenced)
b. Central control (command/response)
c. Distributed control (polling)
d. Distributed control (contention)

The central control, time-referenced system uses a central clock source that provides frame timing to each user. Therefore, each user transmits during a unique slot allocated to it and receives data during some or all other slots. The central control (command/response) system uses a central controller to activate each user by a specific user identification. Thus, both transmit and receive data flow can occur on an individual basis under the control of the central device. Two basic distributed approaches are also available. The first, polling, uses the same approach for data transactions as the central command/response system, while using multiple controllers in a sequential fashion to accomplish control. The second approach to distributed control is contention, which forces each user to

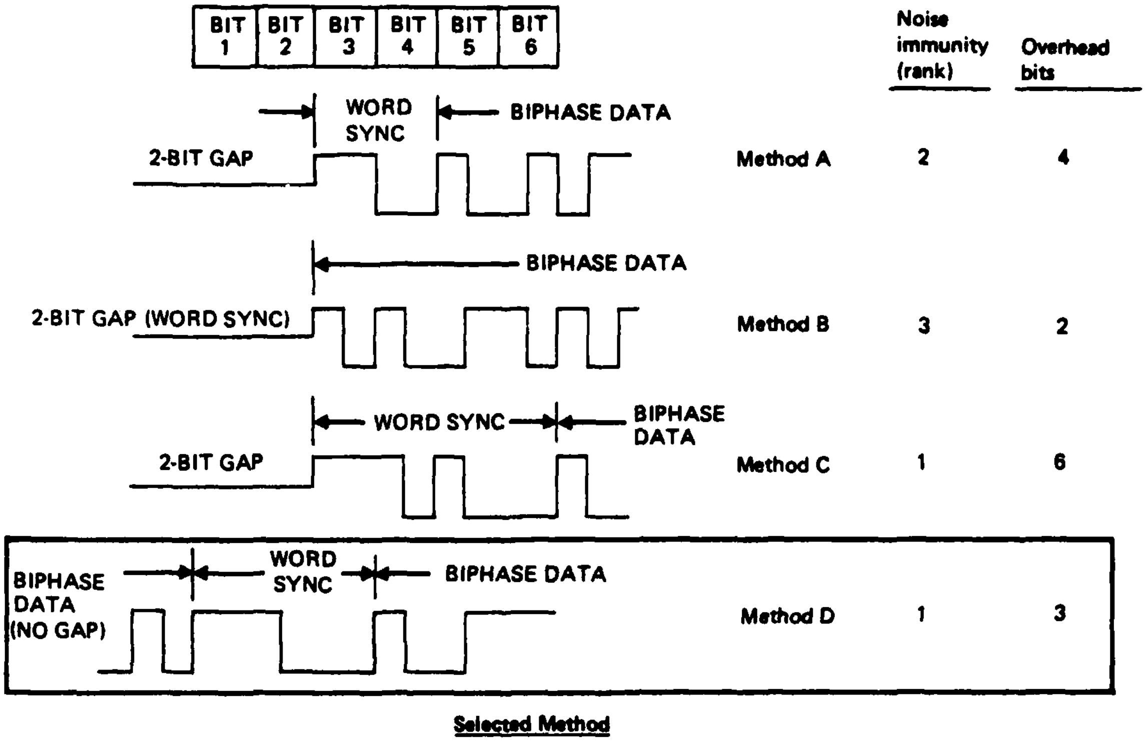


Figure 2-2. Description of Work Synchronization Waveforms

capture usage of the data bus before message transmission is allowed. The bus access control concept that was selected was a combination of both central and distributed control. By choosing a command/response mechanization and then allowing control to be passed to any potential controller (dynamic bus control), both aspects were provided. This approach (command/response) allows the flexibility to meet changing data flow requirements without the restriction of the time-referenced system. The selection of a central control concept also provided the simplest hardware method. The centralized or limited distributed command/response technique permits modifications to be performed in a limited number of devices. Also, system synchronization is easily maintained with limited control devices.

Message Formatting

Each individual word is a string of symbols or parameters in binary form used to convey one or more of the following types of information:

a. Synchronization (bit, word, message, or frame)

b. Supervisory instructions (address of user, data identification, and data quantity)

c. Data

d. Error information

All four types of information can be contained in a single message transaction or may be sent as separate messages. The approach to convey this information is influenced by the message routing techniques and the channelization approach employed. The following items were involved in this selection of message formatting:

a. Data Packing. Selection of a fixed word length simplifies hardware and avoids the necessity of transmitting additional information to specify the number of bits in each word.

b. Data Message Generation. When large numbers of signals originate from the same general physical location and have a common sampling rate, it is often convenient to send these data in messages (block of data words). This approach requires less supervisory information, thus increasing data transmission efficiency.

c. Message Update Rate. Selection of a fixed update rate for each message simplifies hardware and software. The selected update rate determines message latency. Therefore, the update rates may be adjusted to achieve system latency requirements.

d. Supervisory Information. Selection of a format for supervisory information that conveys word synchronization data, user identification, and message identification is essential to hardware implementation.

e. Message Routing. Selection of necessary message types is required to identify system message-routing capability. There are four basic types: user to central controller, central controller to user, user to user, and transmitter (either user or central) to multiple users (broadcast).

f. Bit Error Detection Techniques. The selection of single bit parity on a word basis was found to be sufficient for the basic data bus system operation. This was in contrast to elaborate coding for error control provided by other techniques that produced hardware complexity and data bus overhead; however, the capability to add the more complex error correcting techniques to a unique message has been preserved.

Channeling Methods

Several methods for interconnection (topology) and control of messages were potential candidates for the data bus system. These can be categorized into the types shown in figure 2-3 and are as follows:

a. Single cable using half-duplex transmission method (selected method)

b. Separate supervisory and data cables

c. Supervisory and centrally originated data sent via cable 1 and other data sent via cable 2 (terminal to terminal permitted over data cable only)

d. Separate transmit and receive cables with transmission through central (no direct terminal to terminal provided)

e. Separate supervisory cables to various multiple destinations, with data on a single cable that interfaces with all destinations

The chief advantage of the first method is that it requires less cable than the rest. The chief disadvantage is that it requires greater bandwidth. The third method was not considered further in the comparison because this method is similar to method two and requires additional routing to data registers. Method two also provides better distribution of traffic between two cables than method three.

The key to the selection of the appropriate channeling method involves --

a. Economy of channel bandwidth
b. Flexibility in the programming of the central controller
c. Economy of hardware

Economy of channel bandwidth can be detemined by the minimum data rate required to convey the information. It was determined that if this could be met with the single cable using half-duplex transmission, it would provide optimum hardware economy. This approach also allows the flexibility required by the central controller. Based on several application studies, it was determined that a 1 MHz (1M bps) data bus system would satisfy all communication requirements (control and data).

2.6.2 Signaling Methods and Signal-Detection Techniques

Seven signaling methods were analyzed to determine the technique that provided the best performance in the presence of noise, as well as suitability to specific applications. Key elements affecting signal transmission techniques were filters, transmission media, and coupling transformers. Optimum and suboptimum techniques for detecting the signals were then described and analyzed. The description of signal detection

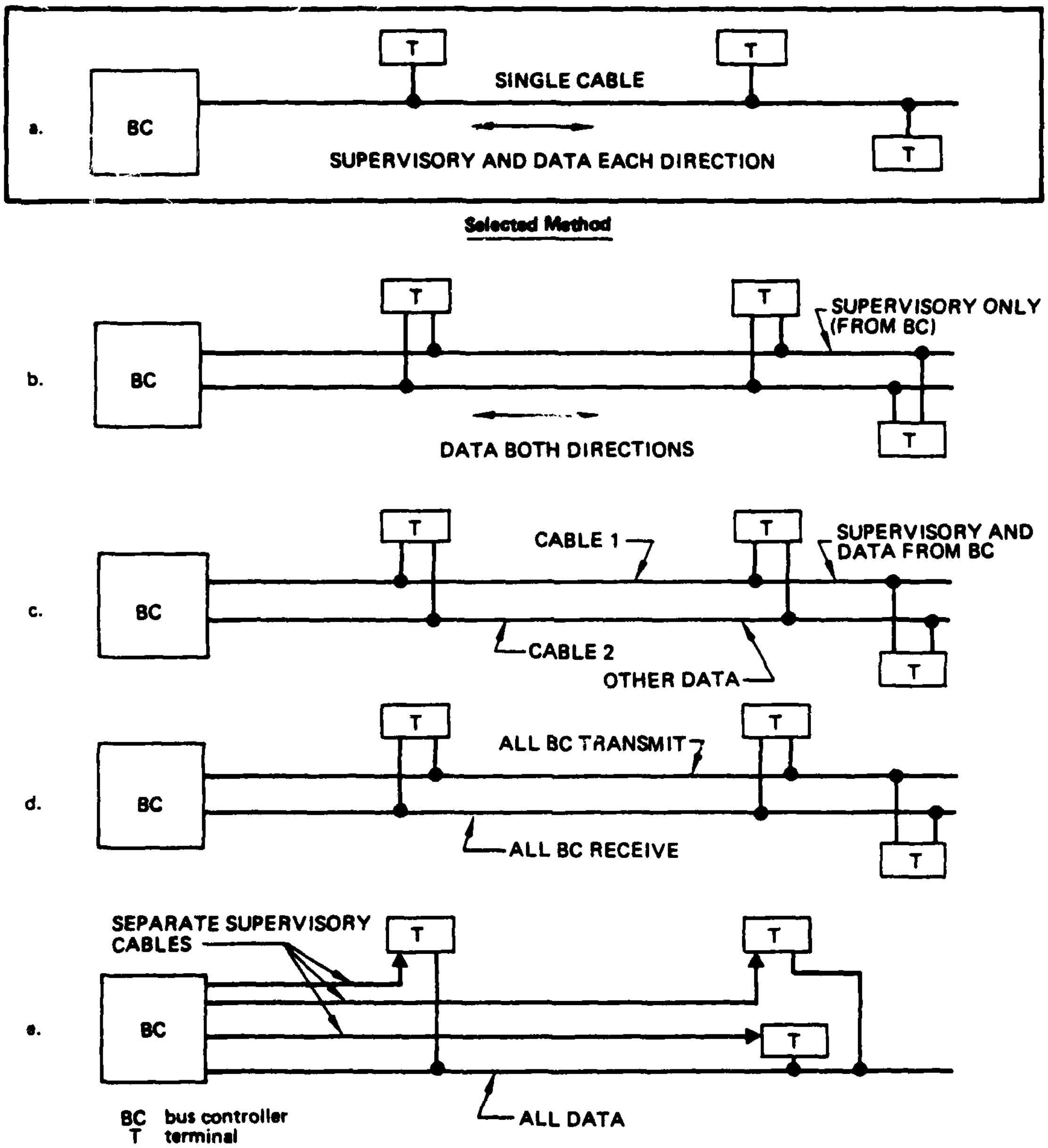


Figure 2-3. Diagrams of Channeling Methods

included a treatment of transmitting and receiving filters, since the effects of filtering cannot be divorced from the detection process. In addition, each signal-detection technique was defined and mathematical models of the noise expected on the data bus were developed to estimate receiver bit error rates. Error detection coding techniques were investigated in relationship to the signal techniques studied. The seven signaling methods investigated were --

a. Unipolar NRZ -- Level
b. Polar NRZ -- Level
c. Polar RZ
d. Biphase-level (selected method)
e. Bipolar NRZ
f. Delay modulation
g. Duobinary

Figure 2-4 illustrates the fundamental approach used to generate each signal. In this figure, minimum transmission bandwidth (B) is expressed in terms of the signallng rate (f_s) in bps. The signaling rate is the highest frequency that must be transmitted to ensure that the message can theoretically be received. The results of the analysis (table 2-1) yielded the following conclusions about signaling techniques.

Unipolar NRZ

Unipolar NRZ is the inherent form in which data are customarily expressed in logic circuits. Because it is not compatible with transformer coupling, it is not suitable for use on the primary transmission medium of the data bus; because it requires twice as much signal power for a given peak-to-peak signal excursion as polar NRZ, it is less attractive for use on short direct-coupled local buses than polar NRZ, which otherwise has identical properties.

Polar NRZ

Polar NRZ is easily generated from unipolar NRZ by subtracting out its dc level. For this reason, it is less suited for internal logic operations than unipolar NRZ. It is better suited for applications on short direct-coupled local buses than unipolar NRZ because it requires only half as much signal power for the same peak-to-peak signal excursion. It is not well suited for use on the primary transmission medium because it is not compatible with transformer coupling. Since polar NRZ does not contain adequate bit synchronization information, it is useful on direct-coupled short local buses only in those applications where bit synchronization is conveyed by some other means, such as via a separate channel. In this case, it is superior to polar RZ because it is less complex and requires only half the transmission bandwidth. It is also superior to duobinary because it is far simpler and because bandwidth conservation is not a serious problem when the cable length is short.

Polar RZ

Polar RZ is superior to duobinary and unipolar or polar NRZ in direct-coupled local bus applications, which require that bit

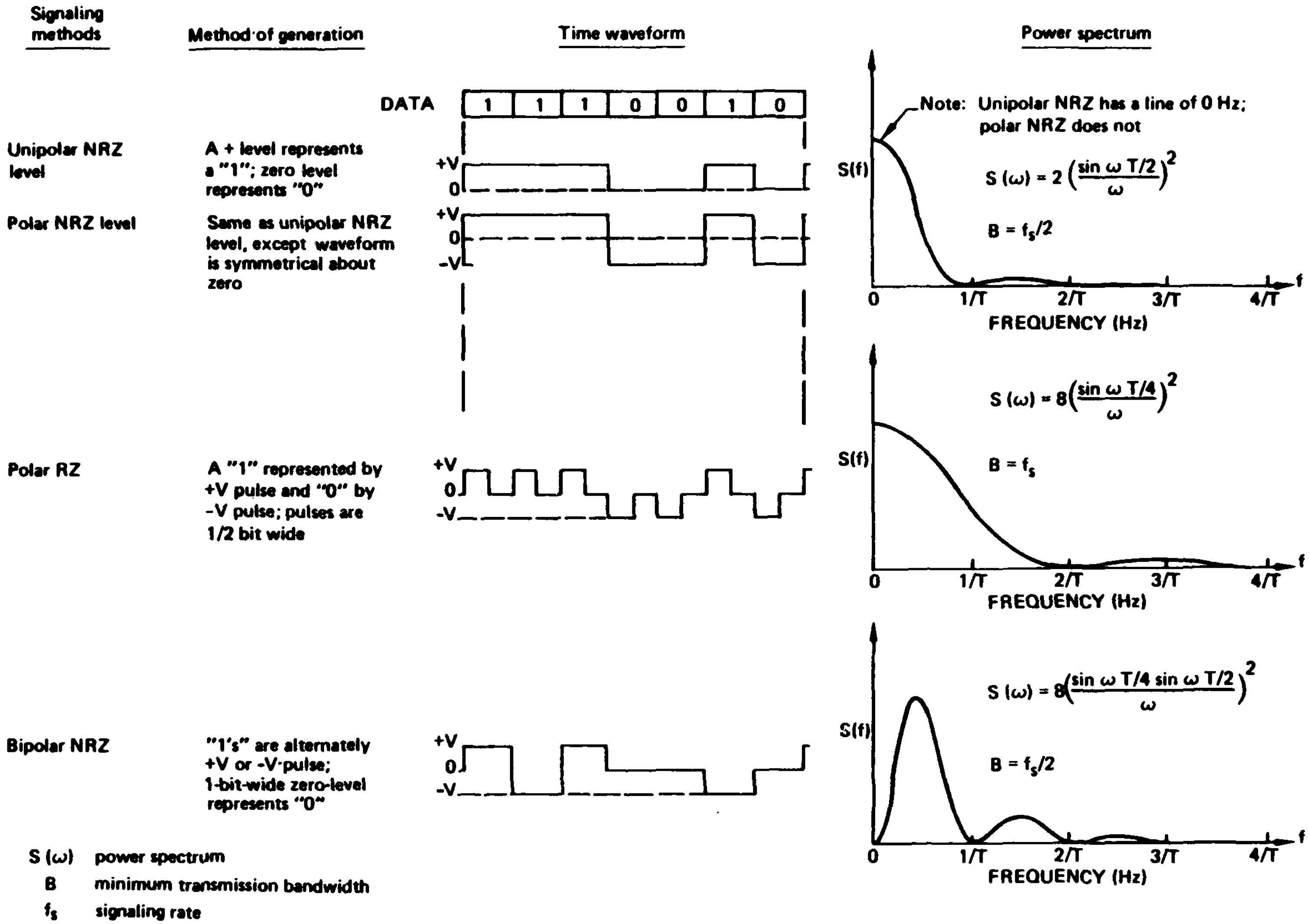


Figure 2-4. Description of Signaling Methods

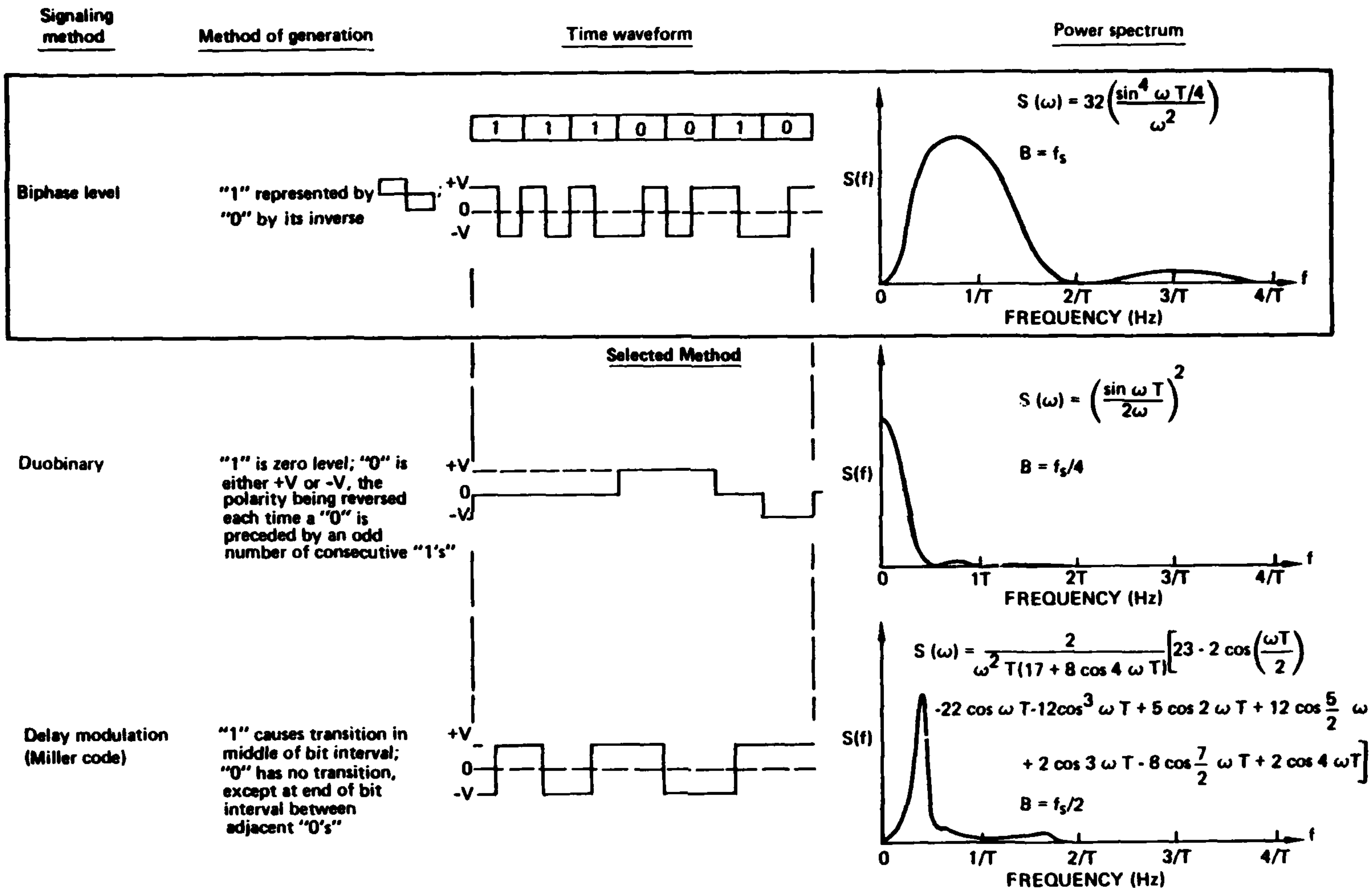


Figure 2-4. Description of Signaling Methods (Continued)

Table 2-1. Summary Comparison of Modulation Techniques

	Unipolar NRZ	Polar NRZ	Polar RZ	Biphase-level (selected method)	Bipolar NRZ	Delay modulation	Duobinary
Inherent error detection capability	No	No	Yes	Yes	Yes	Yes	Yes
Hardware complexity (0 to 5)	0	1	2	3	4	5	5
Required transmission bandwidth/bps	0.5	0.5	1.0	1.0	0.5	0.5	0.25
Compatible with transformer coupling?	No	No	No	Yes	Yes	Yes	No
Self-clocking data	No	No	Yes	Yes	No	Yes	No
Complexity of bit sync circuitry (0 to 5)	N/A	N/A	1	3	N/A	5	N/A
Required SNR (dB for BER of 10^{-6}), peak signal power and noise power	16.6	10.6	13.6	10.6	16.6	16.6	15.7

synchronization information be contained in the basic signaling waveform. While biphase-level and delay modulation could also be used in this application, they are more complex and therefore less appropriate. Polar RZ is not appropriate for use on the primary transmission medium because it is not compatible with transformer coupling.

Biphase-Level (Selected Method)

Like polar RZ, biphase-level consists of a self-clocking waveform and is well suited to applications in which bit synchronization cannot be conveyed by other means. Unlike polar RZ, biphase-level is compatible with transformer coupling and may therefore be used to convey data via the primary transmission medium. In this capacity it is superior to bipolar NRZ for cases where synchronization information cannot be conveyed by a separate channel. Because of its greater complexity, biphase-level is less appropriate than polar RZ for short direct-coupled local buses that require a self-clocking waveform; however, for local buses that require transformer coupling and a self-clocking waveform, biphase-level is markedly superior to delay modulation, which is more complex and requires more signal power. Biphase-level is most appropriate for use on short transformer-coupled local buses or on the transformer-coupled primary transmission medium (which may be long, around 500 ft), provided bit synchronization information must be conveyed by the signaling waveform and cannot be provided via a separate channel.

Bipolar NRZ

Bipolar NRZ, because of its greater complexity, is less appropriate than polar NRZ on short direct-coupled local buses. It is inappropriate on short direct-coupled buses, which require that synchronization information be

conveyed as an integral part of the signaling waveform. Bipolar NRZ may be more appropriate than biphase-level on the primary transmission medium in cases where a separate clock channel is available and bandwidth is a significant factor and/or the error detection capability of bipolar NRZ can be effectively used.

Delay Modulation

Delay modulation, because of its greater complexity, is less appropriate than polar NRZ on short direct-coupled local buses that do not require that clock information be extracted from the signaling waveform. Short direct-coupled local buses that require a self-clocking waveform are better served by polar RZ than by delay modulation, as polar RZ is also less complex. Short transformer-coupled local buses are better accommodated by biphase-level because of its greater simplicity and because there is no significant bandwidth restriction if the transmission medium is short. Because of its greater complexity, delay modulation is less suitable for use on the primary transmission medium than bipolar NRZ (only for applications that do not require a self-clocking waveform). For applications requiring extraction of clock information from the signaling waveform, delay modulation is superior to biphase-Level only if the transmission medium has a bandwidth restriction that prevents effective reception of biphase-level.

Duobinary

Duobinary is not appropriate for applications that require transformer coupling and hence is not appropriate for use on the primary transmission medium. It is more appropriate for applications that employ direct coupling, in which there is bandwidth restriction that prevents use of polar NRZ. This situation does not occur on short local buses.

2.6.3 Transmission Media Considerations

Of the widely varying types of transmission media investigated, it was concluded that a wire cable would provide adequate transfer data rates and provide sufficient reliability, size, and weight characteristics in the aircraft environment. Optical data transmission media could easily replace wire cable in the future. Wire cable transmission media are available in a wide range of forms. The basic types of cable that were candidates for the data bus transmission media included --

a. Twisted-shielded pair (transmission media selected)
b. Twin-axial
c. Coaxial
d. Planar-parallel
e. Triaxial
f. Shielded-balanced pair (four-conductor)

Based on cable tests, twisted-shielded pair was selected as the appropriate cable for the data bus system transmission media.
Several different methods of operating and interfacing with a data bus transmission network were identified as follows:

a. Matched. The impedance into any terminal is matched to the characteristic impedance of the line.

b. Matched and Loaded. The impedance into any terminal is usually greater than the characteristic impedance of the line.

c. Lossy. A network is introduced into each terminal interface to deliberately produce loss into the transmission line to provide isolation of line faults. Usually the transmission line is looped when operating in this configuration, or two transmission lines are coupled together through lossy networks to provide multiple signal paths when faults occur in the transmission line.

<u>After considerable testing of matched and loaded and lossy lines, the matched and loaded line, often called a lossless line, was selected for use.</u>

Four approaches for coupling black boxes to the transmission media were investigated. The techniques examined included:

a. Photocoupled devices to couple the received data to the receiver (multiple transmitters cannot be coupled using this technique)

b. Transformer coupling -- coupling of transmitters and receivers to the transmission line via an isolation transformer

c. Direct coupling -- coupling of transmitters and receivers directly to the transmission line directly

d. Capacitive coupling -- coupling of transmitters and receivers to the transmission line via a capacitor

<u>The transformer-coupling approach was selected</u> because of the advantages it provided: isolation, relatively inexpensive, suitable for relatively long lines, high common mode rejection, impedance transformation, and dc component rejection. 1553B allows both direct- and transformer-coupled approaches for connection to the transmission line, but even in the direct-coupled approach, the black box contains transformer coupling. Therefore, the transmission line is isolated from the user via a transformer-coupling technique.

CHAPTER 3

SYSTEM DESIGN

TABLE OF CONTENTS

LIST OF FIGURES

1553B Figures

LIST OF TABLES

3.0 SYSTEM DESIGN

The emphasis in this section is on the system thinking and system perspective that must be taken to achieve a complete multiplex system design. The section is not a tutorial on system engineering. For that perspective, readers are referred to MIL-STD-499, "Engineering Management," and other textbooks. All topics in the section are directly related to 1553 multiplexing. Many topics of avionic system engineering are not in this section; for example, avionic subsystem hardware selection and mission performance analyses relating to accuracy and capability. Therefore, some general prerequisite knowledge is needed to get each of the topics in an avionic system engineering perspective. Generally this prerequisite knowledge falls into the folllowing areas:

a. General knowledge of military avionic subsystems (sensors, displays, computers)

b. General knowledge of system engineering and system integration activities as practiced by the Air Force, Navy, and airframe contractors

Particular detailed knowledge of avionic subsystem characteristics, such as range, accuracy, use, etc., is not a prerequisite for this section.

This section has five major topics, of which topic 2, "Avionics Integration Design Activities," is by far the longest and most detailed. Topic 1, architecture introduction, is a tutorial, that presents the concept of multiplex system topology and system control as "architecture." In other words, the physical arrangement and connection of a multiplex system is not a complete description of its architecture.

Topic 2, "Avionics Integration Design Activities," contains a mixture of tutorial material, 1553 application advice, examples, and discussion of "issues" when it was inappropriate to give either advice or examples. Eight separate subjects are presented in topic 2:

a. Functional partitioning
b. System control design
c. Establishing an avionic data base
d. Data bus use analysis
e. Bus network modeling
f. Life cycle cost analysis

It is unlikely that a system engineer having assignments in avionics integration would need to be proficient in all of the subject areas, but he or she would need a basic understanding of these areas. The initial paragraphs in each of these subject areas should provide the necessary overview with details being provided in the appendices. The last two subjects describe activities of engineering specialists and are summarized in this section with specifics written with those specialists in mind being provided in the appendices . System control design is a subject that all avionics integration engineers and managers should understand. It is an important subject, and the control system design influences both multiplex terminal design and software.

Two brief, but necessary topics at the same level of integration design activities, follow that topic:

a. Multiplex system requirements and design specification
b. Multiplex system test

Some problems and practices of 1553 multiplexing and solutions are presented in these topics.

The final topic is intended to give some concluding insight into the relationship of engineering management in relation to events in a typical system acquisition and to what 1553 multiplex engineering milestones should be reached.

3.1 SYSTEM TOPOLOGY AND SYSTEM CONTROL INTRODUCTION

3.1.1 Data Bus Topology

Data bus topology is the map of physical connections of the data bus terminals to the data bus. It includes all terminals and data buses involved in the data bus integration of the vehicle. The types of data bus topologies can be categorized into the following two general categories:

a. Single level
b. Multiple level

A single level bus topology is the simplest bus topology and is exemplified by the F-16 avionic bus architecture (see fig. 3.1-1). In a single-level bus topology, all terminals are interconnected via the same data bus. The redundancy requirements of a particular application may require a single-level topology to be implemented using multiple interconnecting cables operating in various modes (active or passive). However, the requirement to use multiple buses for redundancy purposes does not change the single-level bus topology definition if the following criteria are maintained:

a. All terminals are connected to each data bus cable
b. Communication on each data bus is identical

The methods of bus control and the redundancy communication techniques used are peculiar to the application and will be discussed under these areas.

The multiple-level bus topology is an expansion of the single-level topology and can be expressed in two basic forms:

a. Multiple levels of buses with equivalent levels of control
b. Multiple levels of buses with hierarchical levels of control

The multiple-level bus topology with equivalent levels of control is exemplified by the weapon system, that uses multiple, single-level bus topologies for different functions. An example of this relationship is the B-1 EMUX, AMUX, and CITS (see fig. 3.1-2), that represents three, single-level bus topologies with interconnections for data exchange purposes only, thus producing a multiple-level bus topology for the vehicle. Each of these single-level bus topologies operate independently of each other with equivalent levels of control. Another method of achieving a multiple-level bus topology within a subsystem integration is exemplified by the B-52 OAS (see fig. 3.1-3) multiple-level bus topology, that is partitioned into two single-level topologies (i.e., control and display versus navigation and

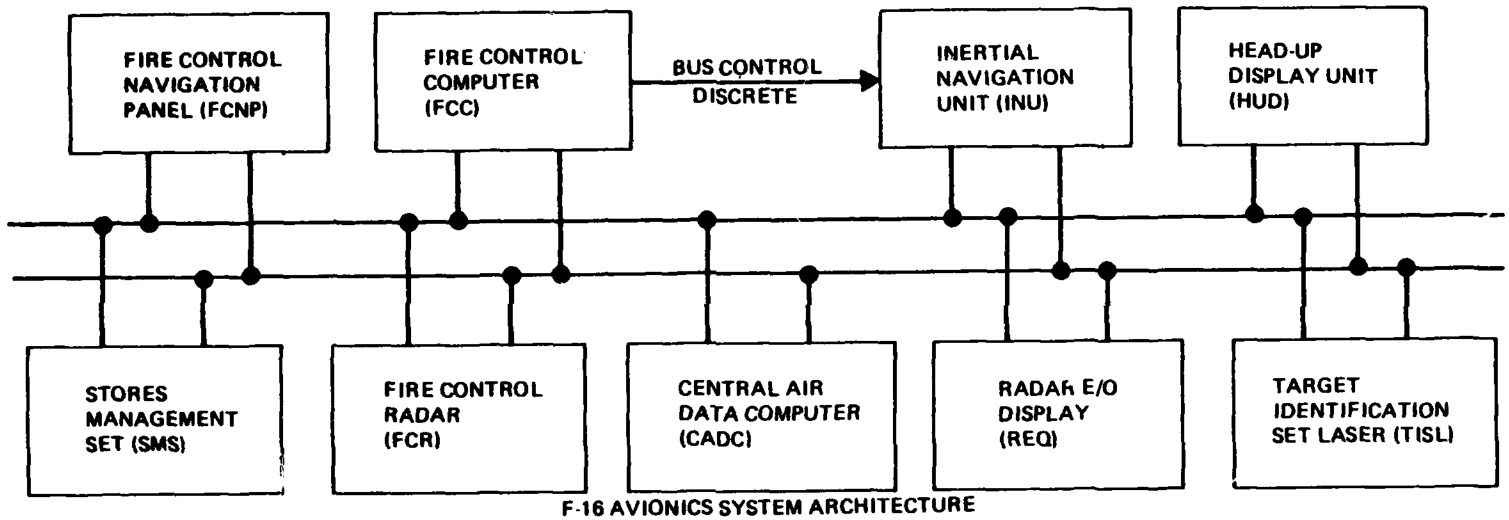


Figure 3.1-1. F-16 Avionic System Architecture

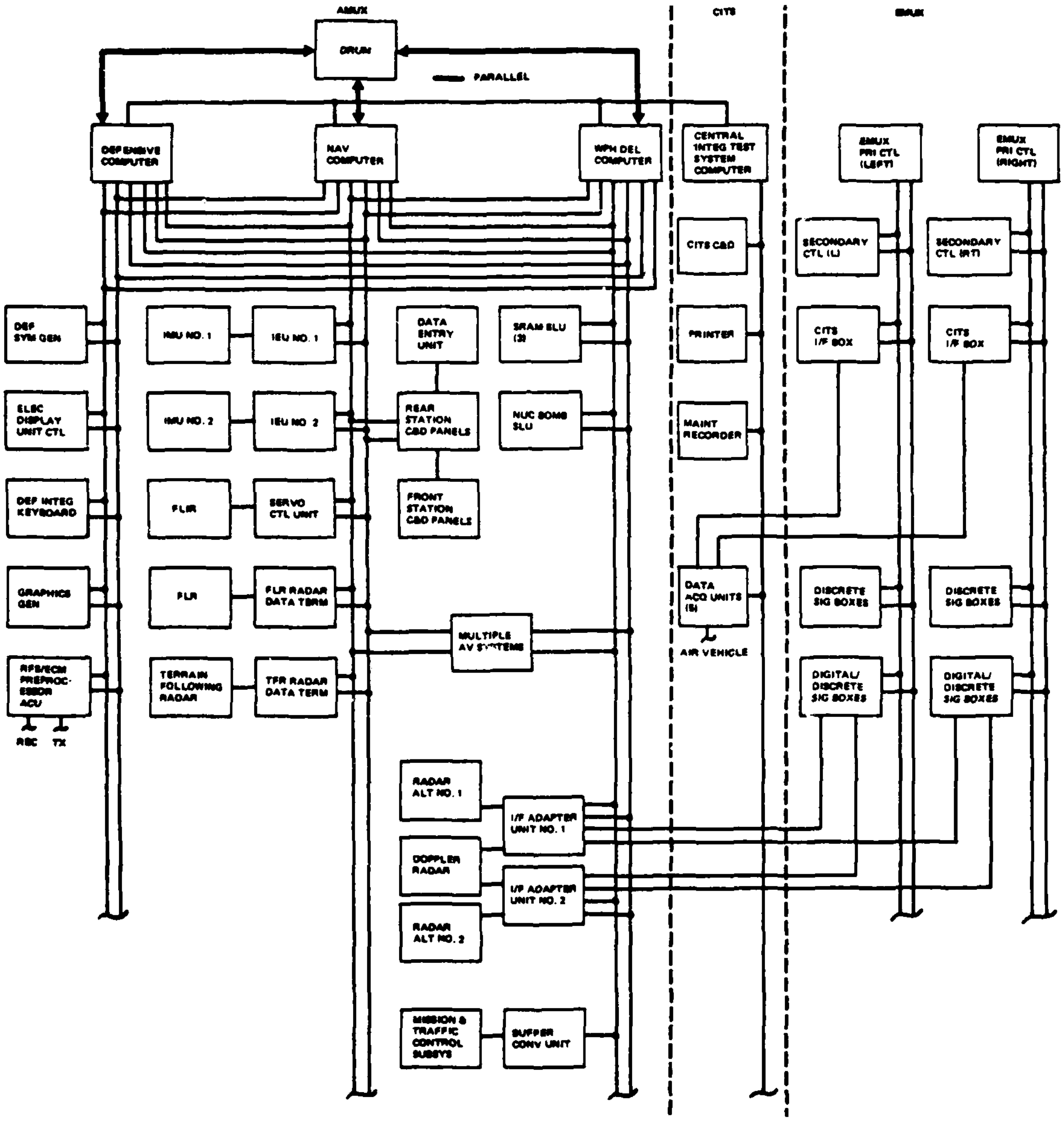


Figure 3.1-2. B-1 Multiplex System Architecture

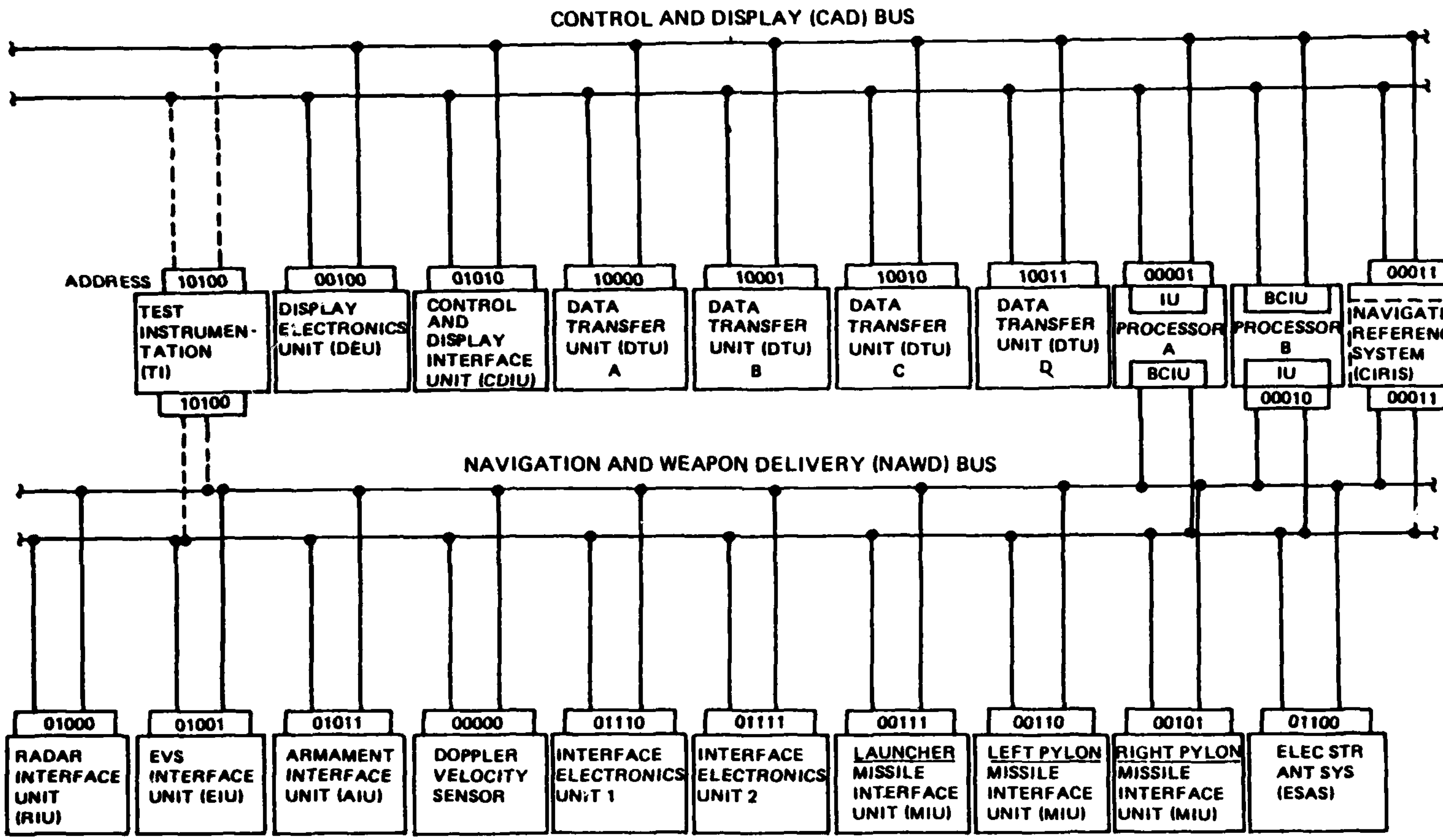


Figure 3.1-3. OAS Multiplex System Architecture

weapon delivery). This trend to multiple, single-level topologies within a vehicle or in a large subsystem integration is a natural evolution of the single-level bus topology.

A second form of multiple-level bus topologies occurs when one or more single-level bus topologies are integrated with another single-level bus topology where the levels have a control relationship (see fig. 3.1-4). The bus level inequality may be expressed as follows:

a. Local buses, subordinate (under submission to global)
b. Global bus, superior (control over local buses)

The primary reason for the difference in control is based on functional usage and not on the interconnection of the terminals. Therefore, data bus topologies can be identified or defined on the basis of interconnection requirements of the user terminals. These requirements also establish the interconnection of a sensor to a local or a global bus.

3.1.2 Data Bus Control

The control philosophy used to maintain the communication on a data bus is described in MIL-STD-1553. The control approaches are of two types:

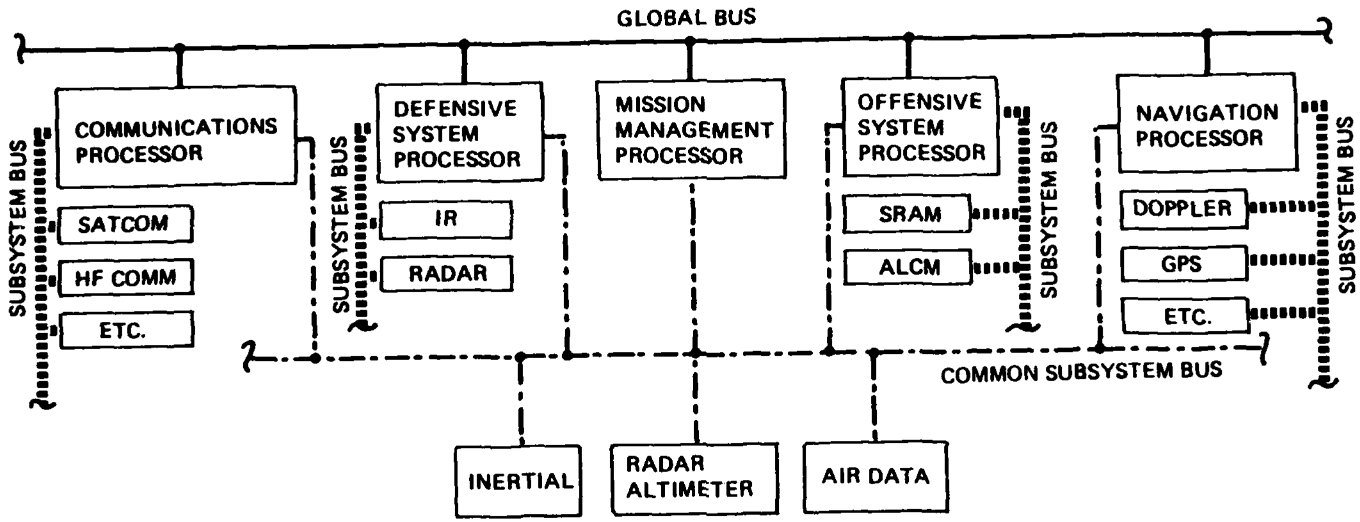
a. Stationary master
b. Nonstationary master

The stationary master bus control concept is used when a single bus controller orchestrates the bus communication for all devices on that data path. Only in the event of a failure of the bus controller hardware or software will another bus controller (backup bus controller) operate the data bus. Obviously, as discussed in the topology section, multiple stationary master bus controllers can exist within a system, each controlling its own data bus.

The nonstationary master bus control concept is used when multiple (more than one) bus controllers orchestrate the bus communication for devices on that data path. MIL-STD-1553B provides a method of transferring control from an active bus controller to a potential bus controller (dynamic bus control mode code). This mode code provides a protocol format for issuing the bus controller offer with the responding status word providing an acceptance or rejection of the offer. Since the military standard prevents the operation of multiple bus controllers simultaneously, a method must be established to determine when the above mode code is to be issued and "to whom" it should be offered. The development of the timing (when) and the ordering (how the selection is achieved) is not specified by the standard and must be established by the system design. Two methods have been discussed in the literature:

a. Round robin
b. Polling

The round robin mechanism uses a fixed listing of bus controller operational order and usually a fixed maximum operating time for each bus controller. If this maximum operating time is maintained by each bus controller, a degree of system synchronism is maintained. Using this approach, each



- Isolated, independent, integrated subsystems
- Independent redundancy and control
- Fault-tolerant reconfiguration
- Standardized bus structure

Figure 3.1-4. Hierarchical Multiplex Architecture

potential bus controller will control the bus during a minor cycle (maximum update rate). Equal time for each potential bus controller is not a requirement. Depending on the application and the minor cycle message traffic, potential bus controllers may require varying bus capacity each time it is in control within a major cycle (minimum update rate). The system engineer should exercise care in this area if user subsystems require synchronous operation. Subsystem synchronization can be maintained under this approach by broadcasting a master clock signal to all users from a single source at a periodic rate. Obviously, with fixed minimum times established for each potential bus controller and with some potential bus controllers having no traffic during some minor cycles, bus bandwidth will be allocated but unused. This will lower the efficiency of the data bus.

Analysis of round robin bus control has shown that subsystem data latency is impacted significantly as the number of potential bus controllers increase (see fig. 3.1-5). Therefore, the advantages of the round robin bus control for global bus level (see fig. 3.1-6) and multiple functions on a single bus level (see fig. 3.1-7) must be contrasted with system performance penalties (i.e., efficiency, data latency, etc.).

The second approach to nonstationary master bus control is the method of polling potential controllers to establish which controller has the greatest need (priority message to transmit) to control the data bus. There are several different approaches using this general technique to pass bus control to the next controller. Since this process is more involved than the round robin bus control transfer, the performance impacts (e.g., efficiency) can be more significant (see fig. 3.1-8). However, since selection of priority is achieved, data latency can be improved (see fig. 3.1-5), and if several terminals collect data in this manner (i.e., asynchronously only when necessary), data transfer requirements could potentially diminish. Therefore, a comprehensive system analysis of each application is essential. In each case, as in the round robin scheme, once the new controller establishes control, performance is identical to the stationary master bus control approach.

3.2 AVIONIC INTEGRATION DESIGN ACTIVITIES

System design begins with the statement of the requirements for avionic functions. The overall statements of hardware and software requirements for a multiplex system will be derived from examination of functions and data requirements in the following areas:

a. Subsystems connected to the multiplex bus (or buses). What is connected to a bus? What are the data paths in the avionic system using the buses? What redundancy of data paths has been provided? What redundancy and/or isolation of function and equipment is required? Answers to these questions provide the overall context of avionics system operation.

b. Missions and modes of each mission. It is necessary to know the complete repertoire of missions, how these missions are supported by functions of the avionic systems, and what particular functions are to be performed during each phase of the mission. These groupings of functions by flight phase are called modes. (Note that these system or sensor modes are not related to MIL-STD-1553 "mode codes.") For

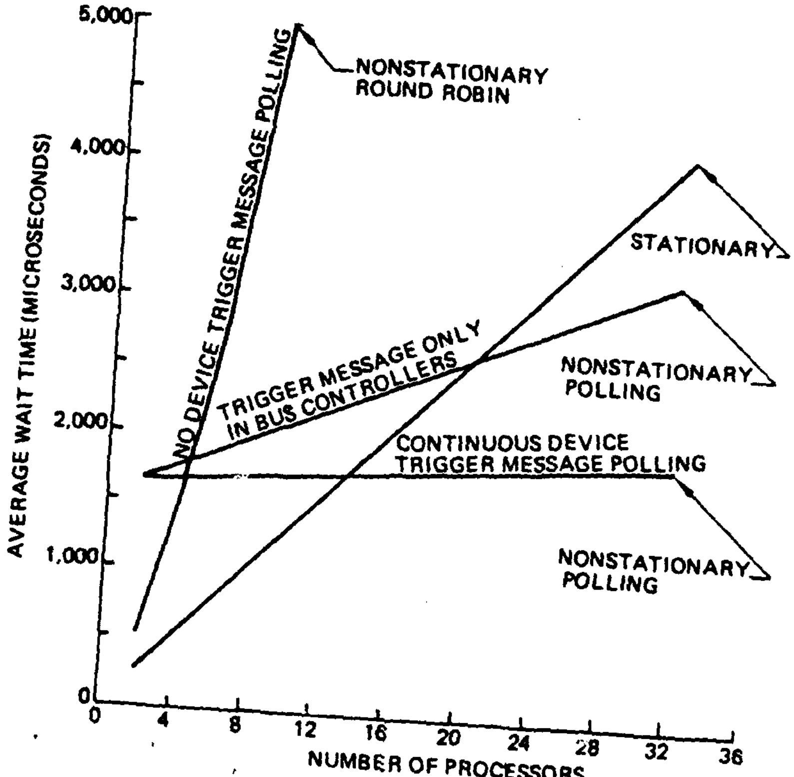


Figure 3.1-5. Data Bus Throughput

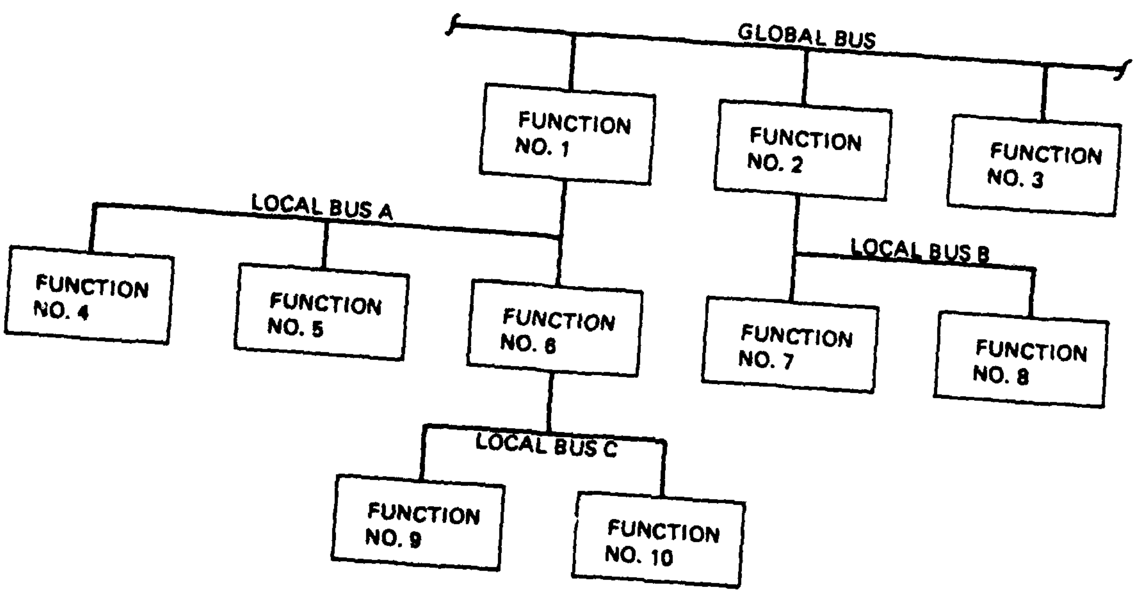


Figure 3.1-6. Hierarchical Network (Multiple Level)

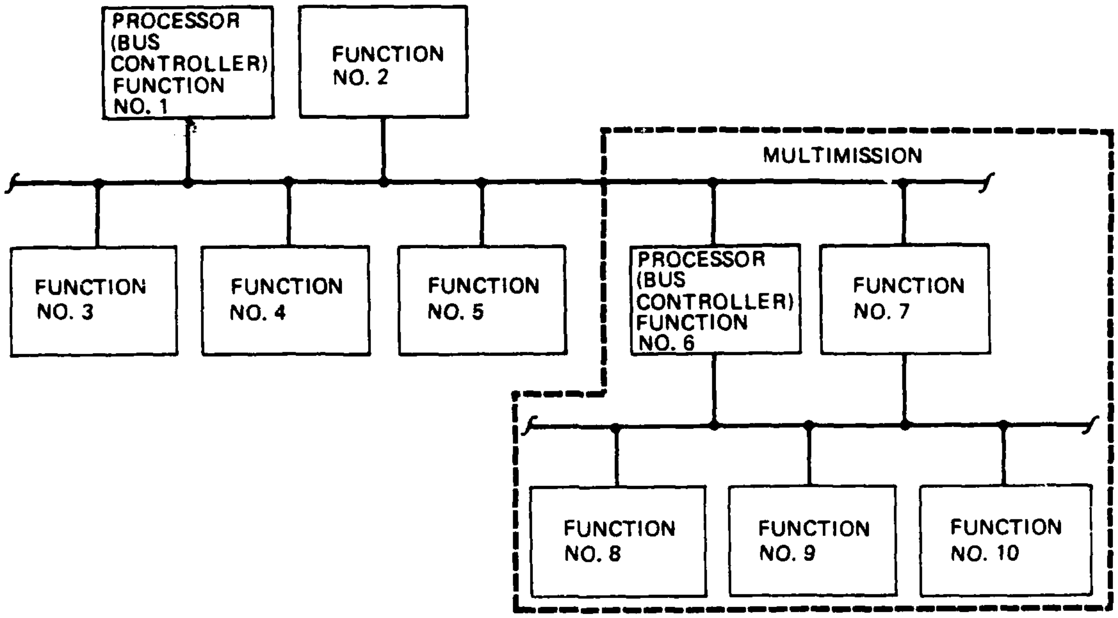


Figure 3.1-7. Single-Level Network

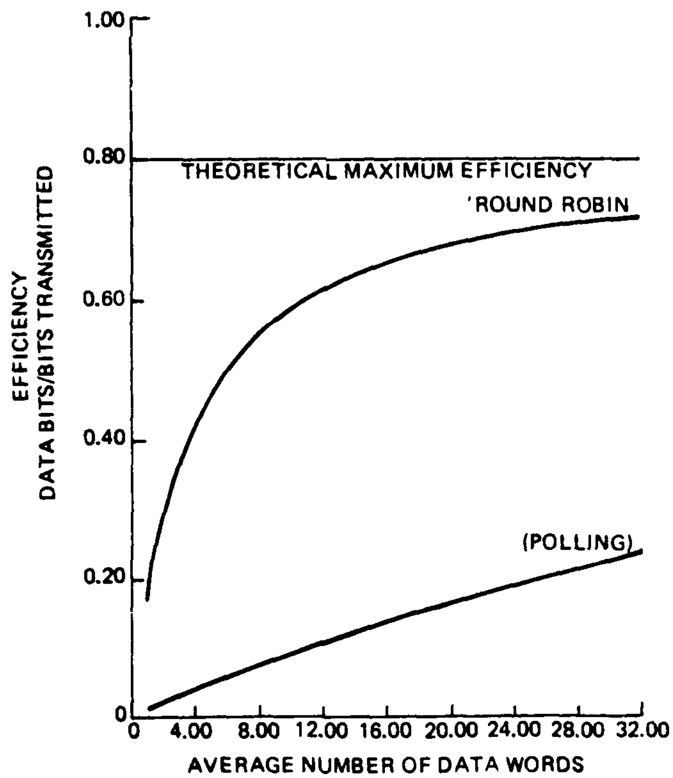


Figure 3.1-8. Data Bus Message Efficiency

example, the weapon delivery mode is usually distinguished from the waypoint navigation mode, even though navigation sensors may be used in each. Apart from the data transfers that are both unique or common to each mode, the unique setup (initialization) conditions must also be known. In short, a functional description of each mission mode, which includes all of the functions of the avionic system, must be known. Requirements that identify time-critical actions or responses must be considered in the development of modes.

c. Functions of each sensor. Descriptions are needed for the inputs that each sensor requires (including sensor control information), what processing (computation) of sensor data is required for all avionic functions, and what data the sensor provides to other avionic systems. The sensor redundancy concepts and how data from redundant sensors will be used or reconciled need to be described, as well as sensor modes versus vehicle (avionic, weapon, flight control) modes. Description of the interrelationship of sensors is required (e.g., inertial navigation update using another position-fixing sensor).

d. Functions of control and display. Descriptions are required for the overall interface of the controls and displays to the avionic systems, as well as which control and display functions depend on multiplexed data.

e. Other avionic functions. The advantages of multiplexing often are applied not only to the integration of sensors, processors, and controls and displays but also to more simple devices like switch positions, actuator positions, and power control. It is because of this application flexibility that the overall use of data in the system must be described.

The descriptions of functions, computations, and modes provided by the answers to b through e above will establish the overall use of the 1553 data bus. It is quite likely that additional dedicated discretes will be used in an integrated system for critical functions (e.g., stores management enable or jettison). These interfaces need to be established and described at the same time the multiplex description is developed.

3.2.1 Functional Partitioning

The redundancy provided by the multiplex system is one of the key concerns and design requirements facing the system designer. It is the system designer's responsibility to obtain the following about each subsystem involved in the integration:

a. The basic level of redundancy within the subsystem

b. The highest mission success probability associated with a function of the subsystem

c. The isolation of the subsystem from other subsystems

d. The independence of the redundant elements within the subsystem

Based on this information about each subsystem being served and any added vehicle particular requirements (e.g., battle damage, no single failure can cause ... etc.), the system designer is able to establish a set of multiplex system requirements. These requirements will impact topology, multiplex control, and avionic control. The topology is usually the most visible point in the multiplex system where redundancy can be observed. However, the system designer must consider much more than just having a topology that meets the observable redundancy requirements. The redundancy of the bus controller and its associated circuitry involved in the detection and correction of a failure as well as the same functions in a remote terminal are all part of meeting the redundancy requirements of the integration.

Before itemizing each of these multiplex system elements with respect to redundancy, certain basic redundancy and isolation issues need to be discussed in general terms. Redundancy is defined as the mechanism used to accomplish a function when the primary mechanism for accomplishing that particular function is not operational. The level or number of potential devices capable of accomplishing the function is identified as the level of redundancy. Another key word used in this field is isolation. Isolation is separation of two or more things so there are minimum or zero reactions of one based on the actions of the other. It should be recognized that the addition of any interconnect scheme required for integration introduces new devices that were not previously involved in the nonintegrated system. Therefore, the action of adding devices to accomplish integration will affect the reliability and isolation of the subsystem to some extent, regardless of their reliability, mission success probability, or the isolation they maintained.

Current multiplex systems exhibit wide variations in redundancy and the method used in the design to achieve redundancy. Basically, three generic elements of the multiplex system are discussed:

a. Data path
b. Bus control
c. Remote terminal

Functional partitioning is achieved in a data bus integration by the proper selection and distribution of all the data bus elements (buses, terminals, and controllers) and the subsystems (sensors) being integrated. The topology associated with a particular application, the bus control method, the normal and abnormal operation of the sensors, the hardware and software partitioning, and the redundancy requirements are all involved in the development of a successful integration.

As discussed earlier, data bus partitioning is basically a topology selection process based on integration requirements, sensor types, and level of redundancy. The single-level topology most closely aligns itself with single- and dual-redundant sensors where complete isolation of dual sensors is not mandatory. It also is applicable to functional integration (i.e., navigation, weapon delivery, communication, etc.) where dissimilar sensors (i.e., INS, GPS, TACAN, etc.) are integrated to achieve a function or a single purpose (e.g., navigation). Multiple, single-level topologies are used to achieve integration of multiple functions (e.g., B-1 EMUX, AMUX, and CITS as shown in fig. 3.1-2) where isolation requirements prevent their integration in a single-level topology. Often these multiple-level

architectures have equivalent single-level control relationships (e.g., B-52, fig. 3.1-3) or control relationships applicable to hierarchical architectures (see fig. 3.1-4). In addition to these obvious partitionings, some other partitionings may develop based on isolation and redundancy criteria (e.g., flight phase essential and flight critical). Criticality does play an important part in the design of a data bus topology. As an example, the isolation of identical channels of a flight control system provides reliability and independence that could change a single-level topology into a multiple-level, equivalent topology (see fig. 3.2-1). Separation of control and display functions, navigation, weapon delivery, communication, electrical power control, propulsion control, or other subsystems could require a single-level topology to become a multiple-level system.

Within each level of topology, the data bus terminal hardware and the sensor interface to the data bus must consider partitioning. Two concerns are apparent to the system designer immediately.

a. Which data bus should a sensor or terminal be interfaced to?
b. What is the proper level of redundancy for the interfacing hardware?

In single-level topologies, the first concern is not relevant. However, in multiple-level topologies, the sensor placement is extremely important, since the primary reason for using the multiple-level topology is to provide separation and isolation not inherent in the single-level structure. Therefore, careful placement of sensors, considering both similar and dissimilar redundancy, is essential. Without this care the system's performance, isolation, separation, or criticality can be compromised.

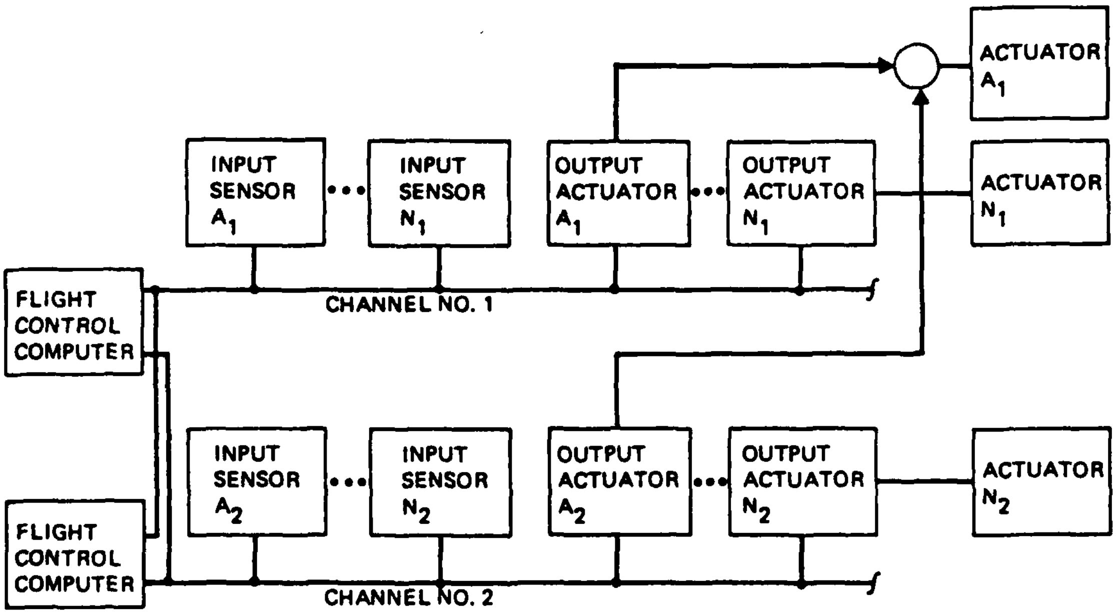


Figure 3.2-1. Typical Dual-Channel Flight Control Bus Architecture

The second concern is the bus interface partitioning. This is applicable to all levels of bus topology. A bus interface can be functionally partitioned in several ways. Figure 3.2-2 shows three methods commonly used today. In the first approach only the analog section is dual. The second approach provides for a dual analog section and a dual digital circuitry through the address decode logic with a common I/O to host interface. The third approach provides independent channels to the host subsystem. Each of these approaches represents a level of partitioning available to the system design when interfacing sensors to the data bus. Remote terminal (RT) hardware, used to interface multiple subsystems to a data bus must face the same functional partitioning considerations. However, in the case of the RT, the bus interface hardware, the internal timing and control section (e.g., a microprocessor in today's world), general-purpose interfaces (e.g., analog-to-digital converters), and subsystem unique interface modules must all consider the functional partitioning.

One of the most important data bus elements is the bus controller. Functional partitioning for this bus control element takes on two forms:

a. Its position in the topology as the communication controller
b. Its internal characteristics as both a hardware interface and a software-firmware system.

It is because of the primary function of the bus controller (i.e., communication control of the data bus traffic) that considerable care should be taken in its application. Understanding of its modes of operation (both the normal and abnormal modes) is essential. Its performance and its robustness with respect to failures are essential to the operational capability of the entire data bus integration. Since bus controller hardware (analog, encode/decode and host interface) can be categorized in the same way as RT hardware, the hardware functional partitioning is similar. However, the remainder of the bus controller partitioning will be discussed in the following section because it involves both the hardware and software aspects of system control design.

3.2.2 System Control Design

There are two separate types of control that must be mechanized for any information transfer system. Control must be defined for the multiplex system, which is responsible for communication of data, and for the avionic system equipment, which creates the input data, processes the data, and displays the results of the computations. Both these control features must be defined by the system engineer, and the resulting design should reflect a separation of these control mechanisms as much as possible.

The following sections will consider that the system functions are known and the control mechanism must be developed from four interrelated points of view: data transfer control, avionic system control, multiplex system control, and bus controller interface hardware.

3.2.2.1 Data Transfer Control

Most multiplexed avionic systems operate on fixed schedules of data transfers. The requirements for the scheduling come from the examination of the largest and smallest minimum iterations and allowable latencies. The

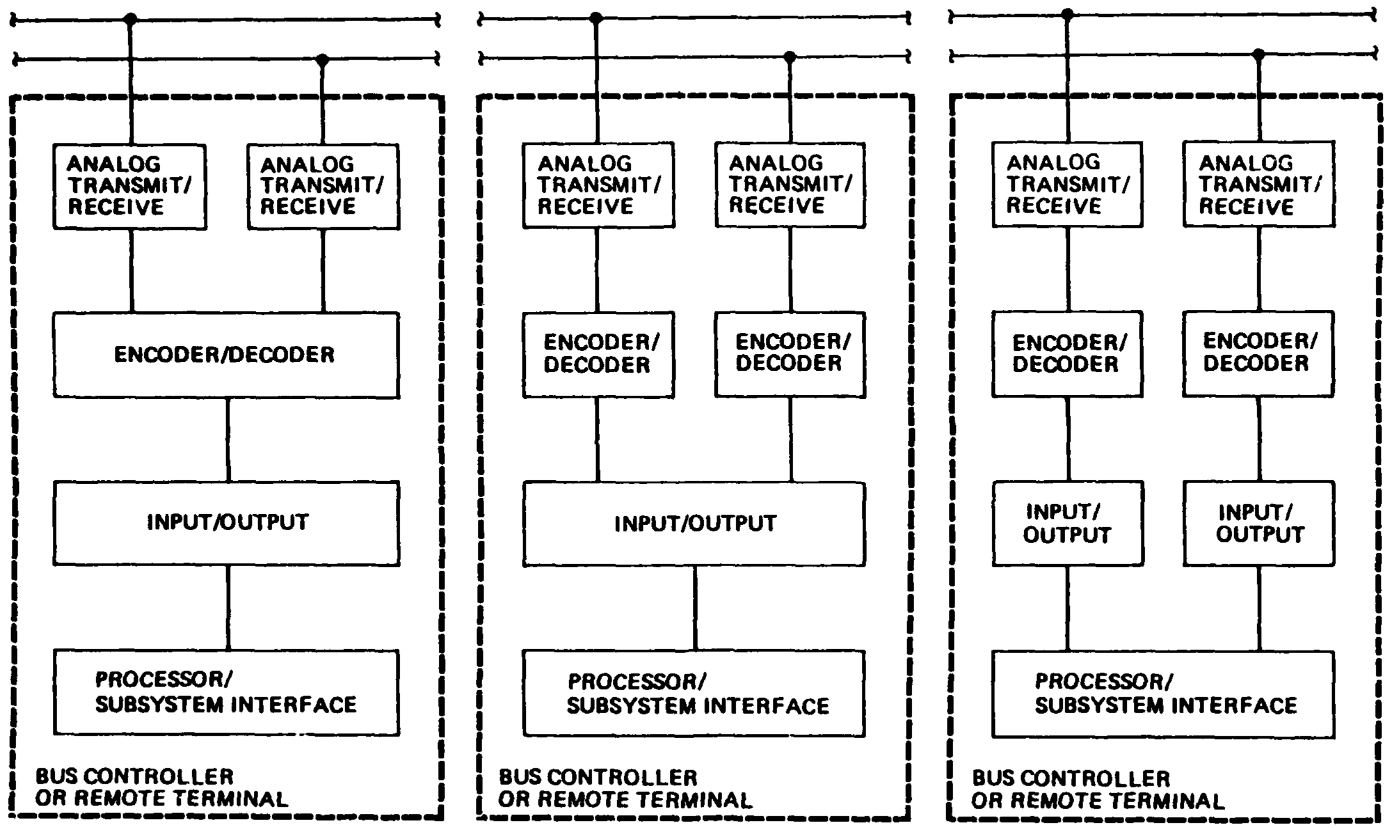


Figure 3.2-2. Terminal Bus Interface Redundancy

slowest iteration rate, which is the least common multiple of the faster iteration rates, is normally defined as the major cycle (see fig. 3.2-3). Over the course of a major cycle, all periodic transmissions occur at least once and all periodic computations occur at least once. Some exceptions do exist if the iteration frequency is very low (such as Kalman filtering once per 6 sec, or periodic built-in-test functions once every 10 sec). The minor cycle is normally the frequency of the most rapidly transmitted periodic data. Typical major frames are 1 sec in length, while minor frame lengths can be binary (2^N/sec) or decimal (10^N/sec) with common values being 1/128, 1/64, 1/50 sec, etc.

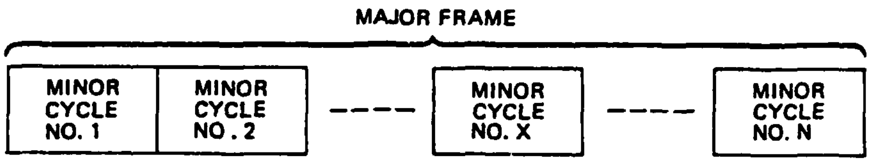


Figure 3.2-3. Major and Minor Cycles

For example, if the major frame is 1 sec long and there are 64 (2^6) minor cycles, then each minor cycle is 1/64 sec or 15.625 ms long. Each periodic message would occur at least once each major frame, up to a maximum of 64 times. If a transaction needed to occur eight times per second, it must occur during one of the first eight minor cycles (64/8 = 8) and every eight minor cycles thereafter. The minor cycle in which the first message occurs is known as the "phase," while the repetition rate is its "period."

In the example of a transaction occurring eight times per second, shown in table 3.2-1, if the first transaction occurred in minor cycle 3 (the phase), later transaction would occur in minor cycles 11 (i.e., 8 + 3), 19, 27, 35, 43, 55, and 63.

Aperiodic messages in avionic systems are rare. They are based on conditional events and are used to initiate other conditional events. The conditional events relating to aperiodic messages might be a requirement to present a display to a crewmember within X-milliseconds of keyed in commands or keyed in requests for data. Another example of a conditional event relative to an aperiodic message might be a requirement to acquire data in a data buffer before it is lost because of new data being inputted in the buffer. This case is typical for keystrokes from keyboards. In addition to nonmachine interfaces, certain data acquisition requirements may be imposed by avionic sensors (weapon delivery, threat warning, etc.). The system designer must know these system data transfer requirements in order to transfer these to the control software, application software, and subsystems that support the required transfers. Close association and cooperation of engineering groups responsible for defining the system functions, system architecture, and data transfer requirements will reduce rework and errors. Once the data communications have been defined and once the functions have been allocated to line-replaceable units, the data can be grouped into messages. There are several practical rules in establishing message contents and scheduling. The analysis tools discussed in section 3.2.4. and

Table 3.2-1. Message Frequency Table

Times per major frame transaction occurs	Period	Possible phases that could occur
1	64	1, 2, 3, . . . 64
2	32	2, 4, 6, . . . 32 . . . 64
4	16	4, 8, 12, 16, . . . 32 . . . 60, 64
8	8	8, 16, 24, 32, . . . 56, 64
16	4	16, 32, 48, 64
32	2	32, 64
64	1	1

appendix B will aid in this process. Several general points can be made about the grouping of data into messages:

a. Do not attempt to group functionally dissimilar information together to minimize the overhead unless necessary.

b. Provide spare capacity in the message sizing and allocations to terminals (maximum message is 32 words and 30 subaddress). Just as functions grow during design and development, so do the communications between functions.

c. Bit packing of data greater than 1 bit should not be done unless necessary. Packing and unpacking takes both time and hardware complexity (e.g., 8-bit analog data should not be packed 2 to a word or discretes packed in with analog data).

d. Attempt to isolate data (functions) that are likely to change over the life of the avionic system from other basic avionic messages to allow for the minimization of disruption of messages because of future modifications.

3.2.2.2 Avionic System Control and Multiplex System Control

MIL-STD-1553 requires that the multiplexed data transfers be initiated and controlled by a bus controller. Each transmission is either a combination of the 1553 protocol and avionic data formatted into 16-bit words or it is a transmission of 1553 protocol without avionic data. The requirements for response time, data rate, and format have imposed requirements on the 1553 terminals that require the bus interface unit to be a distinct piece of hardware possessing certain functions. The system designer must know the specific characteristics of this bus interface hardware design (may differ from sensor to sensor or manufacturer to manufacturer) so that the performance capability of the system can be achieved.

In addition to the normal operation of the system, the system designer must be concerned with bus operation, transmission failures that may occur, the effect on subsystems that are connected to the data bus by a bus controller or remote terminal failure, or the effect of multiplex hardware failure.

Each type of failure will interfere with normal data transfers of the avionic system. The following section discusses requirements that must be defined for certain hardware failures or software errors.

3.2.2.2.1 Avionic Data Transfers and Avionics Control

The types of avionic data transfers that are defined and allowed by 1553B are briefly described below. Each data transfer will also include some multiplex system control information, but that part of the discussion will be deferred to section 3.2.2.2.2. The data found in avionic data transfers will be a combination of avionics control-related actions, such as flags that indicate that a subsystem is initialized, or engineering measurements, such as pitch angle in radians. The multiplex bus is entirely appropriate to effect avionic system control as well as avionic data transfer. Therefore, the normal functioning of the avionic system and the monitoring of its status can and should be accomplished via data bus transfers.

The types of data transfers available to the system designer are as follows:

a. Remote terminal to bus controller. This type of data transfer is used to get data to the bus controller. The bus controller is usually a mission computer, fire control computer, navigation computer, etc. As such it requires data from several sensors such as air data, inertial navigation system or inertial measurement unit, radio navigation, etc., to perform its assigned sensor computational functions. Therefore, its assignment as bus controller is natural. It will initiate the requests to the remote terminals for the data that the processing software needs. The data needs of the mission computer's assigned processing tasks establishes the requirements for data transfers to it from other sources (terminals) on the bus.

b. Bus controller to remote terminal. Typically these types of data transfers are again related to the role of the processor, which has the bus control function. A mission computer may have the requirement to be the data source to sensors, providing such data as position update to an INS, or the requirement to transmit display parameters to a graphics generator. The mission computer often serves as the processor to effect weapon system control, such as fire control, in which case it is controlling both the multiplex system and computing parameters for target designation, weapon initialization, etc. In this case, controller-to-remote terminal transfers are data transfers from the fire control computer to the remote terminals, which contain the interfaces to target designators, stores, etc.

The two types of data transfer described in a and b may also be used as a method of central distribution in which data are taken from a remote terminal to the bus controller and then reformated and retransmitted to other locations.

c. Remote terminal to remote terminal. The bus controller does not need to receive and retransmit all data even though it is in control of the bus. An important class of data transfers is the direct transfer of data from one remote terminal to another, which can be used if the processor that contains the bus controller is not involved in the processing of the data and if reformatting is not required. In avionic

systems that employ more distributed processing (e.g., CADC, INS on the bus), the additional processing capability at those terminals can be used to select and format data for remote terminal-to-remote terminal data transfers.

d. Broadcast. The broadcast data transfer is an option of 1553B, which is not currently in use in military airplane avionics. Broadcast allows the simultaneous transmission of the same data to more than one remote terminal. The broadcast information transfer format may be used for avionic data transfers in the following cases: (1) when significant gains in processing reduction or bus message traffic are needed and (2) when the command/response validation feature of each message is not required. For example, broadcast of roll and pitch data for airplane flightpath control to a dual-, triple-, or quad-redundant flight control system may serve to simplify both avionics and flight control interfaces. The use of broadcast can also be effective when identical data must be transmitted to multiple devices and the latency of serial transmissions will not meet the computational requirements and the command/response message validation feature (per message) is not required. Note that broadcast does not obviate the determination of status (e.g., broadcast message received bit in status word), but that status is not transmitted in response to a broadcasted message. Status can be determined by separate requests of the controller to individual remote terminals to determine their recognition of a previously broadcasted message.

Each unique data transfer must ultimately be identified to the multiplex system hardware and software by its unique combination of terminal address and subaddress. It is this feature that establishes the requirement that the data transfers be organized into messages. Because the decoding of a subaddress (as well as the terminal address) is usually done completely in hardware, the assignment of subaddress for each unique data transfer or data block is normally a system engineer's task. This statement should not be interpreted that the assignment of subaddresses cannot or should not be under bus controller software control. For example, the bus controller software could be used to load a register with a set of subaddresses at the beginning of a particular minor cycle. However, the response time requirements of 1553B (4 to 12 us) will not allow real-time software decoding of each subaddress. Therefore, it is common during the system design process to prepare tables that define the complete data transfer specification by messages for each subsystem on the bus. Entries in the table are usually as follows:

a. Subsystem name, for example, INU, CADC, radar

b. Subsystem terminal address, viz, a 5-bit binary number, per 1553B, paragraph 4.3.3.5.1.2

c. Data block ID, viz, a reference to a detailed word-by-word description of the data

d. Subaddress, viz, a 5-bit binary number per 1553B, paragraph 4.3.3.5.1.4

e. Word count, viz, a 5-bit binary number per 1553B, paragraph 4.3.3.5.1.5

f. Refresh rate, for example, the rate at which the subsystem updates a variable

g. Transmit rate, viz, the intended rate, usually stated as a minimum value, at which the subsystem will be requested to transmit the data

Separate tables are required for transmit and for receive for each terminal, whether it is a remote terminal or a bus controller. If a terminal is a potential bus controller (such as a backup or an alternate) additional tables are required to describe its performance during the bus control mode. The job of the system designer is to define all data blocks that will be transmitted or received under both normal and abnormal system conditions. Obviously, these data will be translated into bus control software; therefore, an exact correspondence between the input and output of data blocks and the use of the data must exist.

3.2.2.2.2 Multiplex System Control

Every data transfer via the 1553B bus contains multiplex system control information in accordance with the 1553 protocol. For the system designer to increase the control capability, two options exist that will affect the hardware and software designs: (1) whether additional data transfers shall be used that are dedicated to the multiplex system management, and (2) how much of the optional multiplex management capability will be used. In addition to the status word, which is routinely received, 1553B provides for "mode control," which according to paragraph 4.3.3.5.1.7 "... shall only be used to communicate with the multiplex bus-related hardware, to assist in the management of information flow, and not to extract data from or feed data to a functional subsystem."

The "types of data transfers" discussed previously are "information transfer formats" according to figures 6 and 7 of 1553B and are described in paragraph 4.3.3.6 of the standard. The definition of the words in the data transfers is in paragraph 4.3.3.5 of the standard. The most common use of these information transfer formats (really message formats) is the command/response formats of figure 6. Note that as a result of using any of these formats, the controller will receive status. Evaluation of the mandatory status word, if received, will establish whether operation of the multiplex system is normal and will indicate that no subsystem failures have been detected.

The evaluation of the status word, if received within the response time of 1553B is divided between the multiplex hardware and software. Nine status bits are available for use, of which two are required and the other seven are optional. Multiplex hardware will evaluate the status word and if none of the flags in the status word are set to logic one, normal operation is ensured. All unused optional status bits are set to logic zero. Although it is the responsibility of the remote terminal to evaluate its own abnormal status and take appropriate action, 1553B does not allow the status word to be reset by the remote terminal independently. Bus controller action is required to reset the previous status. This is accomplished by the reception of a valid command to the RT, which does not request certain mode code data.

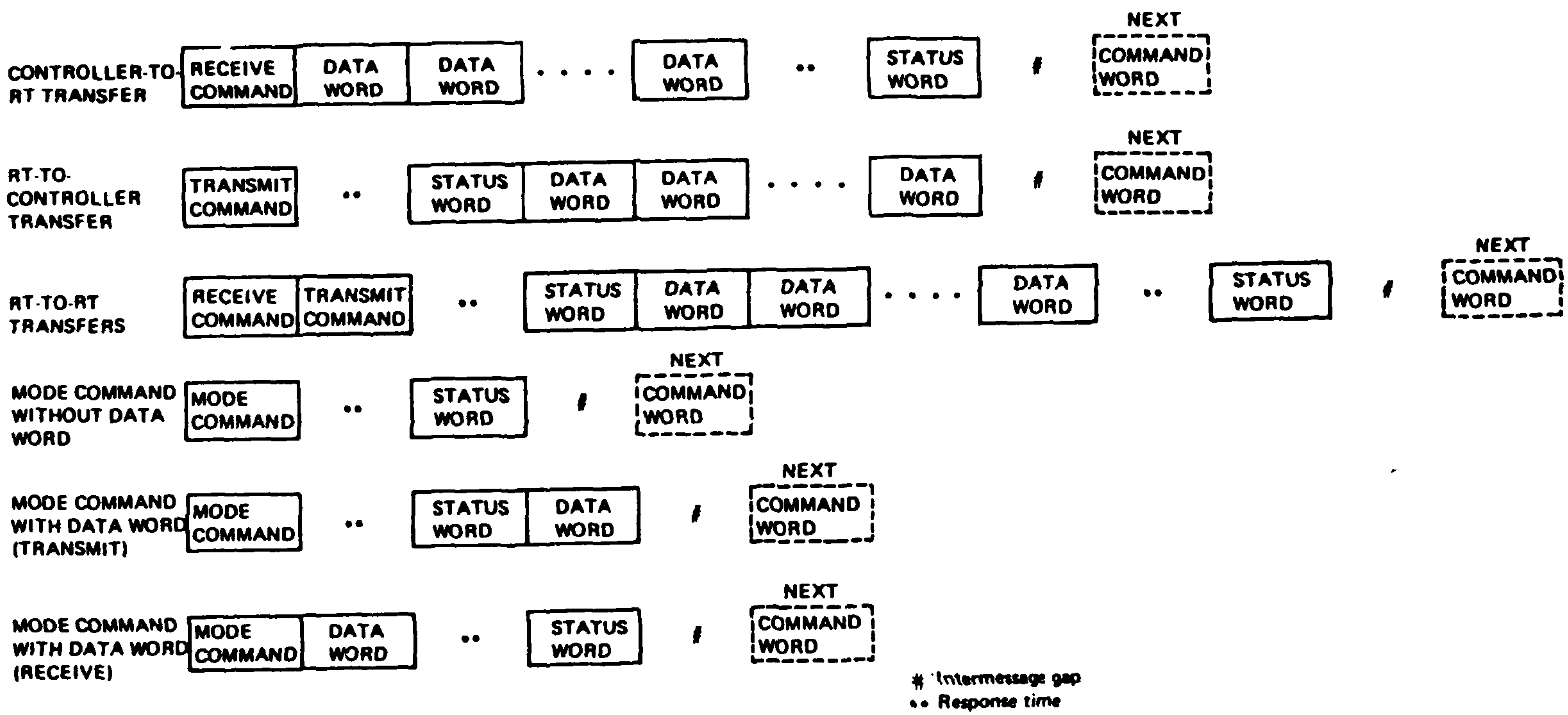


Figure 6 of 1553B. Information Transfer Formats

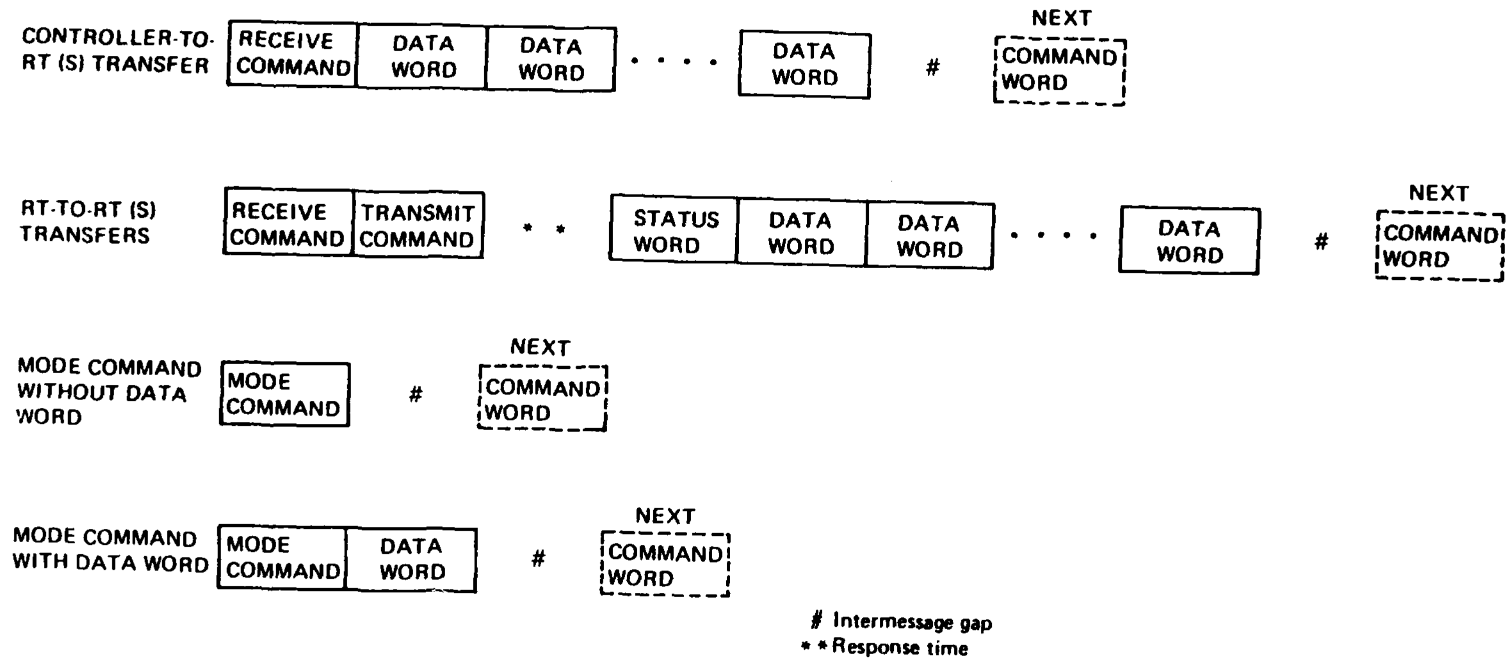


Figure 7 of 1553B. Broadcast Information Transfer Formats

Once the status word is decoded, it will normally be examined by hardware to determine if any of the bits are logic one. The results of finding a logic one usually causes an interrupt so that error detection software can be used to evaluate which status flags have been set. If the status word is not received, hardware will normally indicate nonreception of status word to the software via an interrupt. The system engineer should be aware that certain decisions (like those of interrupt) regarding bus controller operation are open to hardware and software partitioning and may differ depending on the manufacturer. Therefore, system performance can be affected by the approach chosen to mechanize the bus controller.

The majority of the optional mode codes assist in multiplex and avionic system management if flags in the status word are set or if the status word was not received. The use of the mode codes is quite dependent on the system architecture and the system control that is implemented. The 15 mode code definitions in 1553B, paragraphs 4.3.3.5.1.7.1 through 4.3.3.5.1.7.17, should be reviewed to determine if any are appropriate for a particular multiplex system application. The uses of these mode codes are discussed in the following paragraphs.

3.2.2.2.2.1 Bus Communication Control Mechanization

The basic philosophy of the information transfer system is that it operates as a transparent communication link. Obviously, the information transfer system requires management that introduces overhead into the transmission of data. The command words, status words, status word gaps, and message gaps provide this capability. Within the command word, the mode codes provide this management capability. The mode codes have been divided into two groups: mode codes without a data word (00000 to 01111) and mode codes with a data word (10000 to 11111). The use of bit 15 in the command word to identify the two types was provided to aid in the decoding process. Also, the use of a single data word instead of multiple data words was adopted to simplify the mode circuitry. Generally, with these two types of mode commands, all management requirements of an information transfer system can be met.

Each of these mode codes will be discussed with respect to the bus communication control. The first four mode codes are used during normal system operation and the remainder are primarily used for error management of the information transfer system. The system designer must decide which of these mode codes should be implemented for the control of the information transfer system. Good engineering design practice is defined in table 3.2-2 for each element of the data bus system with respect to information formats and mode codes. This capability will provide the designer the flexibility to provide additional control mechanisms that may be required during the growth of the multiplex system or during modifications that occur over the life of the vehicle.

The following is a brief description of each mode code and its use:

a. Dynamic bus control. The dynamic bus control mode command (00000) is provided to allow the active bus controller a mechanism (using the information transfer system message formats) to offer a potential bus controller (operating as a remote terminal) control of the data bus. This mode command would be the primary way of transferring control from

Table 3.2-2. Bus Element Capabilities

Capability \ Bus element	Bus controller	Minimum standalone RT (dual bus)	Standalone RT (dual bus)	Standalone RT (multiple bus)	Minimum embedded terminal (dual bus)	Embedded terminal (dual bus)	Embedded terminal (multiplebus)	Intelligent device– RT or embedded (dual bus)	Intelligent device– RT or embedded (multiple bus)
1. Information transfer formats									
a. Controller to remote terminal	X	X	X	X	X	X	X	X	X
b. Remote terminal to controller	X	X	X	X	X	X	X	X	X
c. Remote terminal to remote terminal	X	X	X	X	X	X	X	X	X
d. Mode command without data word	X		X	X		X	X	X	X
e. Mode command with data word (transmit)	X		X	X		X	X	X	X
f. Mode command with data word (receive)	X			X			X	X	X
2. Broadcast information transfer formats									
a. Controller to RT(s) transfer	X					X	X	X	X
b. RT to RT(s) transfer	X					X	X	X	X
c. Mode command without data word	X		X	X		X	X	X	X
d. Mode command with data word	X		X	X				X	X
3. Mode codes									
a. Dynamic bus control	X								
b. Synchronize	X		X	X					
c. Transmit status word	X	X	X	X	X	X	X	X	X
d. Initiate self-test	X		X	X					
e. Transmitter shutdown	X		X			X		X	
f. Override transmitter shutdown	X		X			X		X	
g. Inhibit terminal flag bit	X		X	X				X	X
h. Override inhibit terminal flag bit	X		X	X				X	X
i. Reset remote terminal	X		X	X					
j. Transmit vector word	X		X	X				X	X
k. Synchronize	X							X	X
l. Transmit last command	X		X	X				X	X
m. Transmit bit word	X		X	X					
n. Selected transmitter shutdown	X			X					X
o. Override selected transmitter shutdown	X			X					X

one nonstationary master to another. Only the single receiver command request (unique address) is allowed to be issued by the active bus controller. The response to this offering of bus controller is provided by the receiving remote terminal using the dynamic bus control acceptance bit in the status word. Rejection of this request by the remote terminal requires the presently active bus controller to continue offering control to other potential controllers or remain in control. When a remote terminal accepts control of the data bus system by setting the dynamic bus control acceptance bit in the status word, control is relinquished by the presently active bus controller and the potential bus controller begins bus control.

b. Synchronize. This mode code informs the terminal(s) of an event time to allow coordination between the active bus controller and receiving terminals. Synchronization information may be implicit in the command word (mode code 00001) or a data word (mode code 10001) may follow the command word to provide the synchronization information. This synchronization information is frequently a minor cycle number or a systemwide time base. This mode code may be broadcast, which allows simultaneous transmission of the command to all applicable remote terminals.

c. Transmit vector word. The transmit vector word mode code (10000) is associated with the service request bit in the status word and is used to determine specific service being required by the terminal. The service request bit and the transmit vector word are the only means available for the terminal to request the scheduling of an asynchronous message if the terminal has more than one service request. The message format for this single-receiver operation contains a data word associated with the terminal's response.

The next two mode codes to be discussed are used in handling message error conditions.

a. Transmit last command. The transmit last command mode code (10010) is used in the error handling and recovery process to determine the last valid command received by the terminal, except for this mode code. Also this mode code will not change the state of the status word. The message format associated with the single receiver last command word contains a data word from the responding terminal. The data word contains the previous 16 bits of the last valid command word received. Notice that this mode command will not alter the state of the receiving terminal's status word, thus allowing this mode command to be used in error handling and recovery operation without affecting the status word, which can have added error data.

b. Transmit status word. The status word associated with mode code (00010) contains the following information:

a. Transmitting terminal address
b. Message error bit
c. Instrumentation bit
d. Service request bit
e. Broadcast command receive bit

f. Busy bit
g. Subsystem flag bit
h. Terminal flag bit

The status word is the normal means by which the bus controller acquires updates as to the functioning of the data transfers and the equipment that affect the data transfers. The use of each of these status bits is discussed below.

a. Message error bit. The message error bit is set to logic one to indicate that one or more of the data words associated with the preceding received message has failed to pass the message validity test. The message validity requirements are:

1. Word validation. Word begins with valid sync, Manchester II code correctly received, 16 data bits plus parity and word parity, odd

2. Contiguous words within a message

3. Address validation. Matches unique terminal address or broadcast address

4. Illegal command. A terminal with the illegal command detection circuitry detects an illegal command

The status word will be transmitted according to the 1553 message formats, if the message validity requirements are met. When a message error occurs in a message format, the message error bit will be set in the status word and the status response withheld.

b. Instrumentation bit. The instrumentation bit in the status field is set to distinguish the status word from the command word. Since the sync field (3 bits) is used to distinguish the command and status words from a data word, a mechanism to distinguish command and status is provided by the instrumentation bit. By setting this bit to logic zero for all conditions and setting the same bit position in the command word to a logic one, the command and status words are identifiable. If used, this approach reduces the possible subaddress in the command word to 15 and requires subaddress 31 (11111) to be used to identify mode commands (both 31 and 32 are allowed). If not used for this purpose, the bit will remain set to logic zero in the status word for all conditions.

c. Service request bit. The service request bit is provided to indicate to the active bus controller that a remote terminal requests service. When this bit in the status word is set to logic one, the active bus controller uses a mode command (transmit vector word) to identify the specific request, if the terminal has more than one service request.

d. Broadcast command receive bit. The broadcast command receive bit is set to logic one when the preceding valid command word was a broadcast command (address 31). Since broadcast message formats require the receiving remote terminals to suppress their status words, the broadcast command receive bit is set to identify that the command was received properly. The broadcast command receive bit will be reset when the next valid command is received by the remote terminal, unless the next valid command is transmit status word or transmit last command.

e. Busy bit. The busy bit in the status word is set to logic one to indicate to the active bus controller that the remote terminal is unable to move data to or from the subsystem in compliance with the bus controller's command. A busy condition can exist within a remote terminal at any time, causing it to be nonresponsive to a command to send data or unable to receive data. This condition can exist for all message formats (control and data). In each case, except the broadcast message formats, the active bus controller will determine the busy condition upon status response. In the case of the broadcast message formats, this information will not be known unless the receiving terminals are polled after the broadcast message requesting their status. If the status word has the broadcast receive bit set, and the busy bit is not set then the message was received.

f. Subsystem flag bit. The subsystem flag bit is provided to indicate to the active bus controller that an embedded subsystem fault condition exists and that data being requested from the subsystem may be invalid. The subsystem flag may be set in any transmitted status word as part of the message format (control and data). In standalone remote terminals, which may interface with multiple subsystems, the subsystem flag bit is logically OR'ed to form a single status bit input. The investigation of the invalid subsystem will be accomplished by the active bus controller using normal message communication to establish the necessary course of action. It is important to note that this analysis by the bus controller cannot be accomplished using mode command (e.g., transmit BIT word).

g. Dynamic bus control acceptance bit. This bit is provided to indicate the acceptance of the bus controller's offer by the active bus controller to become the next bus controller. The offer of bus control occurs when the presently active bus controller has completed its established message list or if allocated time has been exhausted and it issues a dynamic bus control mode command to the remote terminal that is to be the next potential controller. To accept the offer the potential bus controller sets its dynamic bus control acceptance bit in the status word and transmits the status word.

h. Terminal flag bit. The terminal flag bit is set to a logic one to indicate a fault within the remote terminal. Obviously, the terminal flag bit will encompass many more functions in a standalone remote terminal than in an embedded terminal. Generally, embedded terminals will have limited terminal electronics compared to standalone terminals; however, the reporting of terminal failure to the active bus controller is necessary in both cases. This bit is used in connection with three mode code commands:

1. Inhibit T/F flag
2. Override inhibit T/F flag
3. Transmit BIT word

The first two mode code commands deactivate (inhibit terminal flag) and activate (override inhibit terminal flag) the functional operation of the bit. The transmit BIT word mode code command is used to acquire more detailed information about the terminal's failure.

Five mode codes are used to manage terminals because of abnormal operation or as a means of checking each terminal built-in-test circuitry.

a. Initiate self-test. The initiate self-test mode command (00011) is provided to initiate built-in-test (BIT) circuitry within user terminals. The mode code is usually followed, after sufficient time for test completion, by a transmit BIT word mode command yielding the results of the test. The message formats provided to initiate this mode command allows for both individual requests and multiple requests. Notice that the initiate self-test mode command is associated with the multiplex system terminal hardware only. This mode code can be used during system initialization as well as during the recovery procedure after a multiplex system failure.

b. Transmit bit word. The transmit bit word mode command (10011) provides the BIT results available from a terminal as well as the status word. Only the single receiver request is allowed by the active bus controller. The internal contents of the BIT data word are provided to supplement the appropriate bits already available via the status word for complex terminals. Notice that the transmit bit word within the remote terminal shall not be altered by the reception of a transmit last command or transmit status word mode code received by the terminal. This allows error handling and recovery procedures without changing the error data recorded in this word.

c. Reset remote terminal. The reset remote terminal mode code (01000) causes the addressed terminal to reset itself to a power-up initialized state. This mode code may be transmitted to an individual or to multiple terminals. This command may be requested to be issued during system initialization as well as during error recovery.

d. Inhibit terminal flag. The inhibit terminal flag mode code (00110) is used to set the terminal flag bit in the status word to an unfailed condition regardless of the actual state of the terminal being addressed. This mode code is primarily used to prevent continued interrupts to the error handling and recovery system when the failure has been noted and the system reconfigured as required. Commanding this mode code prevents future failure notifications that normally would be reported using the terminal flag in each subsequent status word response. The message format associated with the mode code allows for both single receivers and multiple receivers to respond. No data word is required with this mode code.

e. Override inhibit terminal flag. The override inhibit T/F flag mode command (00111) negates the inhibit function thus allowing the T/F flag bit in the status response to report present condition of the terminal. This mode code can be transmitted by the active bus controller to both single and multiple receivers. There is no data word associated with this mode code.

Four mode code commands are provided to control transmitters associated with terminals in a system. These commands can be sent to a single receiver or broadcasted to multiple users.

a. Transmitter shutdown. This mode code (00100) is used in a dual-redundant bus structure where the command causes the transmitter associated with the redundant bus to terminate transmissions. No data word is provided for this mode.

b. Override transmitter shutdown. This mode code (00101) is used in a dual-redundant bus structure where the command allows the transmitter associated with the redundant bus to transmit when commanded by a normal bus command initiated by the active bus controller. No data word is provided for this mode code.

c. Selected transmitter shutdown. This mode code (10100) is used in a multiple (greater than two) bus structure where the command causes the selected transmitter to terminate transmissions on its bus. A data word is used to identify the selected transmitter.

d. Override selected transmitter shutdown. This mode code (10101) is used in a multiple (greater than two) bus structure where the command allows the selected transmitter to transmit on its bus when commanded by a normal bus command initiated by the active bus controller. A data word is used to identify the selected transmitter.

3.2.2.2.2.2 Bus Controller Mechanization

The stationary master implementation has been the primary focal point in the discussions. The reason for this focus is that the nonstationary master control mechanism is identical to the stationary master with the exception of two areas: (1) transfer of control from one terminal to another and (2) error handling and reconfiguration. These two distinct areas will be discussed in this section.

3.2.2.2.2.2.1 Transfer of Control Example

The stationary master does not relinquish control unless it is unable to perform the normal bus controller functions. In contrast to this the polling nonstationary must relinquish control. The time during which an active controller has control is defined by a maximum length of time that a controller is allowed to be in control or by a maximum number of messages that a controller is allowed to transmit. These control functions could be implemented in either the BIU or the controlling processor. The transfer of control in this example is accomplished using the dynamic bus control mode code and status return. Therefore, the capability to accept bus control must be a part of the BIU because of its setting of the status word bit. The transfer scenario begins when the active bus controller has completed its established priority message list within the allocated system time constraints. Then, the active bus controller transmits a polling request message to all potential bus controllers in a sequential fashion. This message uses a different system-selected subaddress for each potential bus controller that contains the address of the device and the priority of the highest queued message. After the polling request messages are collected from the N potential bus controllers, the active bus controller selects the highest priority requirements, considering its own requirements, and transmits an asynchronous message to the monitor containing the selected bus controller and the results of the responses received from the potential controllers. The active bus controller then transmits the dynamic bus

control mode command to the next controller. The receiving terminal responds with its status word setting the dynamic bus control acceptance bit. Upon receipt of the status word the transmitting unit ceases being the active bus controller and the responding unit takes over active bus control. The new active bus controller transmits an asynchronous message to the monitor identifying itself as the active bus controller. It is the responsibility of the error handling and recovery software in the monitor to examine the data to determine that the selection was accomplished in the proper manner. The previously described transfer of control scenario requires a minimum 100N + 240 us to accomplish, where N is the number of potential bus controllers. The description of the active bus controller allocation procedure is the only additional feature provided by the nonstationary master control scheme. This feature, with its accompanying message prioritizing, adds a completely new capability to the system engineer. No longer does message control initiate from one location, but control is dynamically transferred to multiple locations depending on message priority. The added control overhead associated with multiple active bus controllers must be offset by the advantages of message priority and decentralization of the control site.

The approach to the nonstationary master mechanization can be implemented in two distinct ways:

a. The same BIU as is used in the stationary master can be used in the nonstationary master. This commonality requires that the polling and exchange of control be performed in software.

b. The polling and exchange of control could be performed by the BIU and additional hardware.

The software to control the nonstationary master is equivalent to the stationary master software with the following additions:

a. The capability to perform polling as the master

b. Transfer and acceptance of control as a normal operation

c. The capability to create a polling response that corresponds to the highest priority message ready to be transmitted

d. A message structure or message type that reflects the priority of data so that the polling response can reflect the proper priority

e. Only a single nonstationary master must be responsible for minor cycle updates.

The modules that are affected by the polling of potential bus controllers for transmission priority are as follows:

a. The active bus controller must interrupt the flow of normal message transmissions by completing a message sequence to perform a polling of other controllers, selecting a new controller, and turning control over to it.

b. Each controller will be an active bus controller as in the stationary master, but only one controller will perform timekeeping services and broadcast of minor cycle updates. It therefore is the duty of the level controller (master) to determine that all of the controllers have completed their message transmissions and thus all have the lowest priorities when polled.

c. The priority of a controller needs to be recomputed after each message sequence and upon the generation of a trigger message.

The structure of the message lists would have to be different from the stationary master, and minor cycle update would have additional duties since each minor cycle could imply a new set of priorities and messages to transmit.

The operation of the bus controller polling software would require each controller be assigned a priority subaddress containing the priority of its highest message to be transmitted so that when the current controller performs the polling sequence, each priority is sent to the proper message location in the poller. Once the last device has been polled, the controller is interrupted to perform the priority computation. The first step in the priority computation is to determine the priority of the next message sequence from the controller itself. The priority of the sequence is determined by examining the message sequence list and the sequence counter. This priority is entered into the controller priority (output) message, as well as in the appropriate input priority location for inclusion in the system priority computation for all the controllers. Next, the set of priorities is examined and control is passed to the highest priority controller. If a tie occurs, the control is passed to the highest priority device with an address higher than the present controller. This algorithm should ensure that, by passing control to the next numerical address, each controller will have an equal opportunity to transmit its messages at a given priority level. This approach does require that the message structures be altered from the stationary master and some additional tables added. The message sequence table is a set of static tables that contains the priority of each message sequence to be transmitted, the beginning address of that sequence, and an indication of whether that entry is a pointer to the next message sequence table. Some tables are altered for asynchronous message sequence table, which is dynamically built as asynchronous messages are created that are not trigger messages. Trigger messages receive priority treatment and will go to the head of the transmission queue (via alteration of the BIU next-message register) and alteration of the controller's priority to the trigger priority. At the end of each message sequence is a link to the polling sequence of messages. These messages collect the controller priorities and interrupt the current bus controller when the polling sequence is complete.

The processing of the polling sequence, using a BIU designed for a stationary master sequence of operations, is much more complicated and inefficient than if the polling were done by additional BIU hardware. An alternative design of a BIU or additional hardware would permit the BIU to control the polling and transfer of control and would require no subaddress to be allocated for the polling process. The interface between the BIU and the executive would be essentially the same as the stationary master with the exception that each message would have a priority attached to it. To accomplish this the BIU must perform the following functions:

a. Poll each BIU and determine the highest priority controller. This function could be performed with the aid of a sequence of control messages that the BIU could interpret to determine which devices should be polled and the order of polling (which could potentially simplify the selection algorithm). The priority request would be a mode code and the status response could contain a data word which included a three bit priority code. One desirable feature of this mechanism is that the ITS control and data remain separated in contrast to the first implementation described.

b. Offer the bus control to the highest priority controller, including itself. This process is also done via a mode code and notifies the monitor of the bus selection.

c. A BIU must accept a bus offer and begin processing its message list.

d. The BIU must count the number of words transmitted so that the transmission sequence does not exceed a count that is loaded at BIU initialization. Once the count or the messages are exhausted then BIU must commence a polling sequence. The priority of the BIU that was just in control will be equal to the priority of the next message ready to be transmitted or be the lowest priority, which indicates that it has no messages ready to transmit.

The executive functions need not be concerned with whether or not they are currently in control. The BIU can interrupt when it needs servicing as either an active controller or as a remote controller, and the combination of the interrupt and BIU registers available to the processor can indicate whether the BIU is in the control or remote mode. Asynchronous messages will be attached to the bottom of the transmission list with the exception of trigger messages. Those messages will go to the top of the message list. When the message list is being altered, the BIU should not be allowed to access it. This restriction implies that the BIU should be stopped for a generally synchronously operating system and a semaphore should be used in a generally asynchronous operating system (i.e., whether to use a semaphore to lock the access to the message list or to stop one of the processes accessing the list depends on the frequency of change of the list).

Another form of nonstationary master mechanization is round robin. The round robin transfer scenario may be the same as for the polled scenario, with the exception that the polling process is omitted. The last sequence of message to be transmitted will be the transfer-of-control handshake. The same conditions that apply to stationary and polled nonstationary will apply to the round robin transfer mechanism. However, the discussion of whether to perform the handoff in hardware or software assumes less significance. The control transfer could (and should) be easly handled by hardware. The internal status register of the BIU, should provide bits indicating whether the BIU (1) is able to become the controller, (2) is currently operating as a bus controller, or (3) is commanded to begin operating as the bus controller (for initialization and error recovery).

3.2.2.2.2.2.2 Error Handling and Reconfiguration

The transfer of control creates a new set of error conditions for the nonstationary master that does not exist in the stationary master mechaniza-

tion. All three information transfer systems use several hardware, software, and system error checking procedures to detect errors. The following additional system monitoring checks are required for the nonstationary mechanism to transfer control.

a. Bus controller handover failure. This failure leaves the system with no active or multiple active controllers and occurs when the bus controller allocation procedure is not accomplished as required. This causes the system to stop.

b. Failure of an active bus controller to hand over control. This failure causes the system that is designed for a nonstationary master control scheme to revert to a stationary master. The problem with this failure is that the active bus controller has limited bus control information (for its own communication needs only) so the system operates in a limited (possibly useless) condition.

c. Failure to control (by a terminal with bus control potential). This failure would indicate that a potential bus controller was not accepting or requesting control during a minor cycle when it should have message traffic. The problem with this failure is that the devices under the control of this controller cannot acquire or distribute data so they appear failed.

d. Failure to choose the proper next potential bus controller. The allocation procedure for the selection of the next active bus controller is being violated in this failure mode. The problem with this failure mode is that time-critical messages may not be serviced consistent with their needs.

Each of these detected errors must be identified by the terminal, active bus controller or bus monitor. A discussion of several of these failure mechanisms with example recovery schemes is provided below.

Handover failure occurs when the active bus controller completes its assigned priority message requirements and fails to transfer control to a potential bus controller having a higher priority message list. This failure can be detected by a lack of bus traffic. A distinctive feature of the polling nonstationary master is the almost constant bus usage. Even when no data messages are being transmitted over the bus, the active bus controller is polling the other BIU's to determine if a bus request has been posted. If an extended period of quiet exists on the bus, there may have been a handover failure and the bus lacks a controller. The monitor could for example pursue recovery by transmitting an asynchronous message to the active bus controller, causing it to relinquish control. If it succeeds, control would be offered to the same BIU for a second time. If control is accepted, recovery is complete. Otherwise, the system would be reconfigured. If, during a prespecified length of time, the same BIU again gains control of the bus and fails to transmit, the monitor would reconfigure that BIU out of the system.

3.2.2.2.2.3 Hardware Failures and Software Errors

Provision must be made during system design to handle errors in data transmission, power transients, hardware failures, and data errors. The

1553 data bus, with its prescribed protocol and hardware characteristics, provides superior transmission error detection capability, but the standard allows the system designer to select the remedial course of action to be taken if an error is detected. The 1553 requirements that are applicable to this discussion are:

1553B requirements	Comments
4.4.1.1	Terminal word validation
4.4.1.2	Terminal transmission continuity
4.4.3.1, 4.4.3.3, and 4.4.3.4	Remote terminal operation--acceptance and rejection of commands
4.4.3.5	Remote terminal operation--response of the status word after valid data reception
4.4.3.6	Remote terminal operation--suppression of the status word after invalid data reception
4.5.2.1.2.4 and 4.5.2.2.2.4	Noise rejection--maximum allowable word error rate

Determining the requirement for a response to a detected error is difficult, and no-well accepted guidelines for doing an analysis which shows that one set of responses is superior to another is available. MIL-STD-1553 gives no guidance in this area. Therefore, this is an issue that requires study as well as laboratory investigation (as in a hot bench) to determine both the effect of a response to a cost effective system.

Errors and hardware failures can be classified into reasonably exclusive indications. These classifications, general requirements, and responses are discussed in the following paragraphs, and provide guidance on the implications of responses to detected failures.

3.2.2.2.2.3.1 Detected Message Completion Failures

MIL-STD-1553 defines word and message validation criteria, which were referenced above. If the multiplex terminal hardware detects either an invalid word (1553B, par. 4.4.1.1) or a transmission discontinuity, (1553B, par. 4.4.1.2), the word and message is to be considered invalid. The standard does not specify what hardware characteristic or software process will define that the already received word, words or message is invalid and that it will not be used. Nor is there a mandatory requirement in 1553 that any investigation at all be instituted on detection of an invalid word or message. The common terminal requirements apply to terminal hardware operation as bus controllers, bus monitors, and remote terminals. With respect to a remote terminal, 1553B says: "Any data words(s) associated with a valid receiver command that does not meet the criteria specified in 4.4.1.1 and 4.4.1.2 or an error in the data word count shall cause the remote terminal to set the message error bit in the status word to a logic

one and suppress the transmission of the status word. If a message error has occurred, then the entire message shall be considered invalid." Notice that the requirement is that the entire received message be considered invalid. This message invalidation requirement may cause some systems (i.e., EMUX) a problem. Since the EMUX systems usually have bit-oriented data rather than word or multiple words (message) oriented data, errors in a word following the reception of good data will invalidate good data. It has been proposed that such a system invalidate all data words from the failure to the end of the message and use previously good data words. This approach however, has not been allowed. Regardless of the approach, some system mechanisms will store the data and then tag the message as being invalid, others will not allow the user to receive the data. In the first case, it is the responsibility of the user to examine the message valid indication prior to using the data, however, in the second case, the user must recognize that the data has not been updated. What the above quote says, in effect, is that a remote terminal cannot use any part of a message that has an error. Message completion failures are always detected in a 1553 multiplex system and are known to the bus controller by either the suppression of the status word or the setting of the message error flag in the status word. This message error flag removes ambiguity as to whether the error occurred before the message was validated by the remote terminal or in the response to the message.

Several points need to be made with respect to defining (or finding) the requirements imposed on the system with respect to message completion failures.

a. What indication will the bus controller terminal hardware provide to the software that indicates a transmission failure has occurred? Usually this is initiated by an interrupt, which causes software to examine hardware registers in the bus controller terminal.

b. What automatic retry of the last message does the bus control terminal hardware implement and to what extent is this under software control?

c. What are the consequences of (1) ignoring the lack of message completion; (2) postponing action; (3) retransmitting the same message? This latter question may be important for the case when the message received at the RT was valid (and therefore used) and the message completion failure occurred during the RT transmission of the status word or reception of the status word by the bus controller.

d. What mode code usage has been planned for the avionic system? The 1553 mode codes provide a capability for investigating the details of a message completion failure. Mode code usage is optional, so the system designer needs to know what mode codes are desired for each RT. It is desirable that each of the RT have the same capability of response to mode codes, but because of the availability of different types of hardware and GFE requirements, it is not always possible to do so.

3.2.2.2.2.3.2 Detected Subsystem or 1553 Terminal Failures

Subsystem or 1553 terminal failures may be detected using built-in test circuitry. The 1553 standard makes provision for the reporting of either of these failures by the setting of the subsystem flag bit or the terminal flag

bit in the status word to logic one. (The use of either of these bits is optional.) Generally the data output of a subsystem and the BIT or validity should be provided as a normal output and examined by the software using the data from that subsystem. Another method of determining multiplex system performance is by loop testing. Loop testing can be accomplished within a multiplex system at several levels. One method is for each BIU to examine its own transmission with its receiver and compare results to determine if transmission errors have occurred. Another method is to transmit a command to a remote terminal output circuit and monitor the output with an input circuit of the remote terminal and report the results to the software in the bus controller for comparison.

The requirements for action for this class of failures are more apparent than for message completion failures, because there is no ambiguity as to the type of failure or its location; that is, given a detection of failure, what failure was detected and what action should be taken? Again, several points need to be made with respect to defining the requirements imposed on the system as a result of such a failure.

a. If dual-redundant buses are used, a terminal failure may be isolated to one bus. Depending on the capability of the remote terminal hardware and mode codes implemented, the transmit BIT word mode code can be a powerful diagnostic aid. Note that this mode code may not be used to request subsystem built-in-test results. For each fault, the action to be taken must also be determined, designed for, and implemented by the system.

b. Determining which subsystem failure caused the subsystem flag is more complex because there is no mode code similar to transmit BIT word for subsystems associated with a remote terminal. Polling of the subsystems connected to the terminal and evaluation of the responses may be required.

Subsystem or terminal failures can be detected without the use of the optional terminal or subsystem flags. For example, repeated message completion failures to a remote terminal via all possible data paths could be considered as a loss of the terminal functions. Bad data or nonvarying data from a subsystem may be interpreted as a subsystem failure. System software (as opposed to bus control software) should be used to detect these and other failures.

3.2.2.2.2.3.3 Bus Controller Switchover Failures

Bus controller operation in the event of failure is important to an integrated data bus system. Several methods for switchover to backup bus controller are being applied in military systems today. The key to successful backup bus controller design and implementation is the ability to meet these criteria:

a. Primary bus controller recognizes internal failures and ceases operation.

b. Backup bus controller recognizes the failure of primary controller and initiates action to take over control.

The simplest approach is the bus controller GO/NO GO discrete (see fig. 3.2-5), which is shared between the primary bus controller and the backup bus controller. Periodically (example, 15.625 to 50 ms), the primary bus controller must signal the backup bus controller indicating that it is still in control of the bus and is in good health (i.e., both hardware and software). An extension to the above approach, that can be added to the above is a monitor which monitors the data bus for activity. This approach uses the assumption that extended periods (for example, 120 ms) of no communication by any user means a primary bus controller failure has occurred and the backup bus controller needs to be activated.

Another possible bus control failure results in an active bus controller retaining control of the bus indefinitely. The monitor is capable of detecting this condition because it is monitoring the system. A scenario is presented here in which the monitor corrects this condition. In a polling nonstationary master system, each minor cycle contains at least one polling sequence. During this polling sequence, each potential bus controller is requested its bus control priority and sends the active bus controller a message containing a status word and a data word defining its bus request priority and terminal address. The monitor receives these data in an asynchronous message from the active bus controller and can thus determine when a bus controller is requesting the bus. Once a bus request is posted, bus failure to transfer control during this period indicates either a dominant or an ineffective active bus controller. When the monitor detects this condition, it can wait for the next polling sequence and then set its bus control priority to gain control, causing the active bus controller to either relinquish control or act to gain direct control. If the monitor requests control and this fails, the monitor must obtain control directly (i.e., using an alternate bus and reconfiguring). If the monitor succeeds, control is offered to the highest priority bus controller. If an ineffective active bus controller regains control of the bus and fails to relinquish it, the monitor reconfigures the BIU/PE out of the system.

The monitor must be capable of determining if a bus controller is capable of gaining control of the bus. To achieve this goal, each bus controller should gain bus control at least once each major frame for synchronous

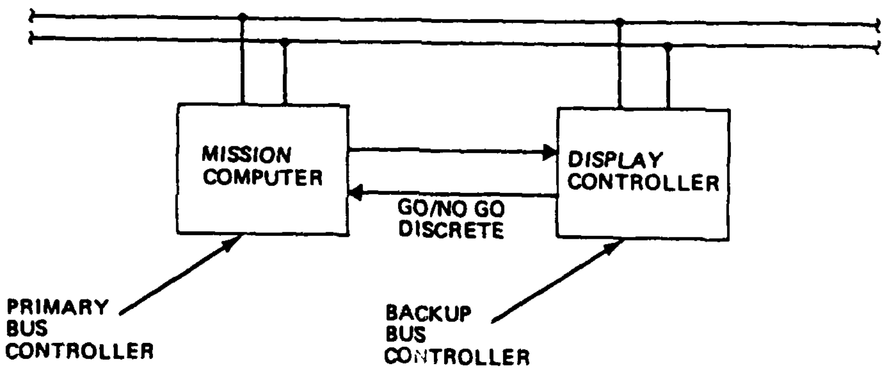


Figure 3.2-5. Bus Controller GO/NO GO Discrete

transmissions. Some potential applications may generate systems containing a bus controller with no synchronous bus (major frame) requirements. If this situation arises, the monitor must examine the health of the system using mode control commands to determine if operation is satisfactory. The monitor is informed as to the cycle each bus controller requires bus control for synchronous transmission. When this transmission is scheduled, the monitor will verify that bus control is transferred to that bus controller and bus transactions occur. If control is not transferred, a quiescent controller is indicated. When the monitor detects this condition, it follows a reconfiguration process.

The stationary master information transfer system normally contains an active bus controller to control the normal activity and a backup bus controller that may either be functioning as a remote terminal performing avionic functions or simply waiting to assume control. All other system failures are handled by the active bus controller. In a nonstationary information transfer system the question arises as to how errors should be handled for communication and subsystems. It is clear that the current active controller should perform error retries to specific devices. If retries fail to accomplish the information transfer, how should the error be handled? Should it be handled immediately? If the error must be handled immediately, should each controller be able to handle all errors? It is doubtful that this should be the case. A controller should be most concerned about data that affects others, while leaving system control to a monitor or reconfiguration device.

One advantage of polling/round robin approaches is that the control of functions and equipment is partitioned into definable entities. A controller causes information to be read from one device and written into another device. The controller must be responsible for the devices it moves data into, but may or may not have any management control over the devices it transfers data from. Therefore, when defining a set of controllers, avionic subsystems, and remote terminals, the data flow should be a primary consideration when defining the partitioning of control. A controller that does not control a particular device could in this case turn the control over to the appropriate device controller to deal with that device's problems. The result of error handling can be broadcast or transmitted as part of any system device status update. The reasoning for this concept is that several bus controllers are multiplexing the control of the data bus rather than using separate buses to perform the same communication function. This case is equivalent to a degenerate hierarchy as shown in figure 3.2-6.

In addition, each of the terminal's fail-safe capabilities required by 1553B (par. 4.4.1.3) prevents incoherent constant transmission by any terminal's hardware. The implication in the system design is that the backup bus controller must recognize this failure. Since the timeout can be reset, (see 1553B par. 4.4.1.3) the analysis of what sequence of actions is appropriate must be done carefully to avoid repetitive resetting of the bus on which the timeout occurred. This case reduces to total loss of function and silence at the end of the timeout period or immediate silence.

3.2.2.2.2.3.4 Detected Data Errors by Software

The 1553 data bus does provide superior error detection capability of

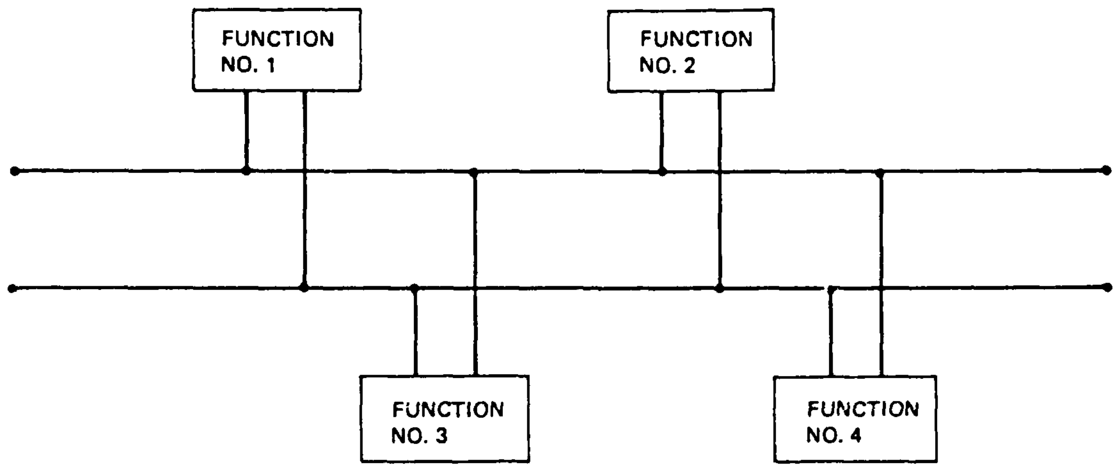


Figure 3.2-6. Hierarchical Network (Single Level)

messages intended to be transmitted and received. This does not mean that inherent errors in data are also detected. Therefore, a systems engineer may need to consider other means of data integrity (data reasonableness checks, check sums, cyclic redundancy check, echo, etc.). These additional data checks beyond reasonableness may be required to provide mission critical success data. Certain methods appropriate for defining detected transmission errors to the software are:

a. Use of "tag" words. MIL-STD-1553 establishes requirements on the terminal hardware design to detect errors in words, message continuity, or message word count. If errors are detected in word count, the message is not to be used. Since validation cannot be completed until the message is completed, hardware designers must make provision either for buffering and discarding an invalid message or establishing a method of tagging invalid data. Tag words are generated by some hardware designs to define the error state of the message. This is an attachment to the received message in memory. Several approaches to accomplish this idea have included the minor cycle number and the number of data words in the tag word along with a validity bit. This allows the applications software to examine the tag word on all data that it has received to determine whether the data are valid. The tag word represents the only indicator of whether the data were received properly and whether they were received during the anticipated minor cycle.

b. Use of error detecting and correcting codes

c. Echo checks of data

d. Multiple copies of critical data items. Techniques b, c, and d are related to data that users consider so important that the very small undetected bit error rate of 1553 (10-12) is not tolerable. These techniques are not unique to 1553 and have been used in many data system implementations.

3.2.2.2.2.3.5 Bus Controller Interface Hardware Mechanization

MIL-STD-1553 allows bus controllers to be implemented as standalone units. An example of this would be a programmed hardware (PROM or ROM) controller, with no significant mission management capability. However, all current designs of bus controllers are of the processor-coupled type. In this type of hardware configuration, the bus controller is linked to a bus control function. This bus control processor normally has additional "mission" processing requirements. The software complexity to implement this bus controller design is totally dependent on the sophistication of the bus controller hardware. That is, the more simple the design, the greater the software interaction required to process a command or message. Examples of bus controller capability are:

a. Single word. This most primitive of processor-coupled bus controllers requires software interaction for each and every word of the message. Though a software routine can be written that specializes in transmitting words to the bus controller, the message processing burden remains in the software.

b. Single message. This type of bus controller has the capability of processing one complete message at a time. The processor software sends the starting memory address of the message to the bus controller, which, in turn, performs a DMA into the processor memory for the required message. The message, including the command word, is completely formatted by the software. The bus transaction is then processed under direct control of the hardware, which signals the processor at the end of message. The software is then left to examine the returned status information (status word, etc.) in order to ensure message completion. In this type of bus controller, the software interaction is lessened by the added capability within the hardware.

c. Multiple message. This type of bus controller features hardware to allow processing of more than one message with a single software action. This means the bus interface hardware can recognize both a normal message completion and all message completion failures. A set of messages are structured and their starting addresses are placed into a table in memory. To initiate processing of these messages, the starting address of the table of addresses is passed to the bus controller. The bus controller commences DMAs into the processor memory, stepping through the table of addresses until either all messages are processed or an interrupt is generated because of an error. This approach allows the software to examine status words as they are returned, and an interrupt is used to identify the completion of the list. Though the software is simplified by the added complexity in the hardware, considerable software activity may still be required for those applications where message structure or table organization must vary during the performance of the bus control function. This type of controller is clearly the most desirable I/O controller for most applications.

3.2.3 Establishing an Avionic Data Base

The data source is an important aspect in the development of a data base, because the data base is made up of existing hardware and software

descriptions that will be used to predict hardware and software of the future. If the system designer waited until all hardware and software designs were completed, there would not be sufficient time remaining for system integtation. Therefore, certain assumptions are required before proceeding with the integration process. Obviously, the idea behind the development of a data base is to limit the assumptions to allow a highly "usable" data base. This data base would be established on a quantitative rather than a qualitative basis. The idea of using well-defined equipment to predict future equipment needs is not a new concept. It has been and will continue to be used by aircraft wiring and installation designers to plan airplane layouts. However, these designers are concerned primarily with wire count and not electrical interface characteristics, but if the same principle that applies to installations could be applied to the electrical interface of equipment, with the proper amount of work a data base could be achieved.

To show that this concept is an acceptable one, a quick review of some basic aircraft subsystems is required. The primary subsystems include--

a. Flight
b. Propulsion
c. Navigation
d. Payload
e. Aircraft systems
f. Defense

A review of modern-technology vehicles will indicate minor changes in the aircraft systems aboard these vehicles. These basic generation and distribution subsystems have seen hardware technology improvements but generally little changes in the interface characteristics from aircraft to aircraft. Basic avionics have remained unchanged functionally and electronically as have other aircraft systems. Few external interface changes have occurred although internal-to-box technology improvements have been significant. One other event that has an impact on the data base is incorporating existing avionics within a single box. This does little to the data base other than eliminate the interbox traffic.

Most flight instruments also come under tight control by military specifications and generally have seen only facial changes, not electrical interface changes over several aircraft models. There have been increases in the number and type of instruments; therefore, the data base should reflect this by listing sufficient hardware items for the modern aircraft. There also has been the evolution to a newer type of displays for basic flight and mission instrumentation (CRT). These instruments integrate many of the functions previously accomplished by individual instruments into a general-purpose (multifunction) display. Generally, these are digital in nature and with proper coordination of standards the digital interfaces will be compatible with the data bus interface. Thus, direct interconnections to the data bus(es) are reasonable and practical. Flight and propulsion controls, which have in the past been exclusively analog, have shown little change over various aircraft models. With the advent of digital flight controls and with a future trend toward fly-by-wire systems accompanied by electronic fuel management and engine inlet controls, the trend toward increased use of digital subsystems in the flight control area is predictable. These subsystems will still contain some analog sensors and

displays, but the type and number will decrease rapidly. System integration of these new digital subsystems will again be complex unless limited numbers of well-defined digital interfaces are established.

Based on this brief discussion about the evolutionary aspect of aircraft subsystems, what can be concluded? Present production hardware and software can be used to simulate future production hardware in the electrical interface area for most aircraft subsystems. Where evolutionary (analog to digital) steps are being considered within a subsystem, new data should be added to the data base to supplement it. This can be accomplished by updating the data base with developmental hardware and software interface characteristics. Therefore, the data base will become a mix of the latest production hardware and software (very-well defined interfaces) and developmental or preproduction hardware and software (reasonably well-defined interfaces). Usually, making this substitution is a fairly simple process because of the functional commonality of the previous sensor with the new sensors.

Generally, a data base can be described using certain key parameters. Table 3.2-3 describes the general data base requirements and how well the Multiplex Systems Simulator (MUXSIM) and Standard Interface Applicability Analysis Program (SIAAP) data bases have conformed to the general requirements.

3.2.3.1 Tools Needed for Data Base Analysis

Two computer programs to assist the designer have been developed under Air Force Avionics Laboratory contracts. Conceptually, these programs only do part of the job the designer has always been required to accomplish. However, through the use of software simulation, the system designer can be relieved from the laborious tasks associated with this work, thereby establishing the design as the decisionmaker rather than data handler. These programs are the Multiplex Systems Simulator (MUXSIM) and the Standard Interface Applicability Analysis Program (SIAAP). The results of the operational evaluation of these software systems yielded positive results. In essence, the operational evaluation proved that these design tools are necessary to adequately investigate architectural alternatives. Descriptions of MUXSIM and SIAAP are presented in appendix A.

3.2.3.2 Benefits of Computer-Aided Design Tools

There are several areas where these two simulations can be useful. In the aircraft industry, three basic groups could be expected to benefit from the data base and methodology established in simulation programs like these. Two of these groups are the technology and design organizations involved in preliminary airplane studies and actual airplane designs using total or partial sensor integration. Detailed aircraft integration information is necessary regardless of the implementation method (digital box-to-box communications or data bus system integration). The third group requiring outputs of these simulation programs is the wire integration designers who need wire count and signal-type information by subsystem. Presently, this group uses existing airplane drawings to retrieve data by subsystem. These data are already available (in more than sufficient detail) from the automated data base developed for these simulation programs. The data base can be sorted by subsystem, hardware unit, or connector. Thus, production

Table 3.2-3. Data Base Requirements Versus Analytical Tools

Data base parameter	Available in SIAAP	Available in MUXSIM	Comments
1. Signal number	Sequence number	Signal ID	Same definition different title
2. Signal name	Yes	Yes	–
3. Origin	Source	Origin LRU code	Same definition different title
4. Destination	Sink	Destination LRU code	Same definition different title
5. Reference number	Drawing number	Configuration	Source of data
6. Connector pin	Source and sink ID	Origin LRU number and pin number and destination LRU number and pin number	Same type
7. Type of signal	Class	Type	Classifies signal by characteristics (i.e., ac analog, dc analog, discrete, synchronous, digital)
8. Signal characteristics		Origin: • Voltage range, parameter range, • Scale factor, resolution, • Quantization, • Parameter rate of change, • Update rate, bit rate, • Signal implication, • Frequency response	

Type	Source
a. Discrete passive load requiring ac (LH), passive load requiring dc (LD), and active load ac or dc (CD)	Logic 1 voltage Logic 1 impedance Logic 0 voltage Logic 0 impedance
b. Synchronous (LS)	Voltage and source impedance
c. Analog (LA)	Voltage range, source impedance, voltage minimum ac or dc
d. Impedance (LZ)	Impedance range, maximum voltage, impedance minimum, and R, L, C, or z
e. Current (LC)	Current range, source impedance, current minimum ac or dc
f. Frequency (F)	Frequency range, frequency minimum, nominal source voltage, nominal source impedance

wire count, wire separation information, connector needs, and signal classification are available in a concise printout.

Since SIAAP and MUXSIM perform similar functions in aiding system integration definitions, it is recognized that the basic SIAAP recordkeeping and electrical analysis can be combined with the MUXSIM recordkeeping and data analysis. Then, the system designer would have the capability to run the full hardware and system integration analysis on a common data base using one recordkeeping procedure. To accomplish this, certain input and output formats would have to be standardized. Such standardization would produce a library of hardware interfaces that would be useful and reusable. Since the military is leaning toward more integrated subsystems, such readily available and accurate definition of GFE hardware interfaces will be essential to accomplish such tasks. Also, an area that has not been fully covered is the software task and interrelationships to the data bus integration.

It is noted that the trend in avionics integration is to standardize on a single electrical interface (MIL-STD-1553), thus reducing this area of interface analysis while increasing attention to the data bus analysis area. However, this is not the case in the integration of other subsystems (electrical, environmental, hydraulic, etc.) where many electrical interfaces via remote terminals will continue to exist. Thus, it is essential that automated tools be available in both of these areas to provide the most detailed analysis possible early in the integration program, thereby eliminating the potential of cost escalation and program schedule delays.

In summary, the following conclusions can be drawn concerning software-aided design tools:

a. They are complementary to manual analysis and can be an aid in system definition.

b. The two programs discussed accomplish the task that they were designed to accomplish.

c. MUXSIM definitely allows the data network design to be formulated and verified and its data traffic control constructed via a fully automatic software tool. The SIAAP program allows the definition of standard interface hardware requirements for remote terminals in an equally automated manner.

d. One drawback is that tools like these must be planned for and made operational prior to the hardware program start.

e. These tools are designed to support integration programs from inception through operational deployment.

f. Any system integration task could be accomplished more economically through use of these tools. A typically large aircraft integration program produces thousands of pages of manually generated documentation that could have been produced using these software design aids.

Therefore, the system designer should pay more attention to and plan for the use of tools like this to aid in the designing of integrated systems.

3.2.4 Data Bus Use Analysis

The use of the data bus resources requires the careful management of this resource. Since both multiplex system overhead (mode control and BIT), data bus message format overhead (command/status words) and avionic system data are involved, an analysis of the bus resources must be started early in the definition stage of a system and maintained throughout the development and acquisition period and well in the post delivery time period. There are some basic analyses which must be accomplished to obtain sufficient visibility of resource use. These include:

a. Bus loading. Actual transmissions as a percentage of the maximum possible (allowable) transmission (both data and overhead are involved in this calculation)

o Efficiency. The ratio of data bits transmitted to the total number of bits transmitted

o Latency. The delay time which occurs from the initialization of an event to the detection and transmission of the event data to its final destination

The data base analysis tools (SIAAP and MUXSIM) are described in appendix A. Appendix B first presents example analyses of three multiplex control mechanisms. The analyses presented illustrate the type of analysis initially required in a multiplex system design effort. The multiplex control mechanisms analyzed are:

a. Stationary master
b. Nonstationary master (polling)
c. Nonstationary master (round robin)

Secondly, appendix B presents results of using data base analysis tools for multiplex system analysis.

3.2.5 Bus Network Modeling

Another important design activity that must be completed early in avionic system integration is the verification of waveform integrity at the receiver of each terminal on the 1553 data bus. This is usually done initially by computer models. Appendix C contains a discussion of the need for models and two example models.

The first simulation model is the AMUX Subsystem Simulation program package, which was developed for the B-1 program (contract F33657-72-C-0600). This model was designed to derive a time-dependent response of a particular AMUX configuration.

The second simulation model is the AFAL Data Bus Network Simulation. This simulation program and related verification hardware was sponsored by the

Air Force Avionics Laboratory and reported in AFAL-TR-75-209 "Data Bus Network Simulation." This general-purpose digital computer program provides the capability to aid in the design and evaluation of complex cable networks.

3.2.6 Life Cycle Cost (LCC) Analysis

The life cycle costs (LCC) of a multiplex system embrace both acquisition costs and operating and support (O&S) costs. LCC analyses are performed early in the acquisition phase to help the designer make decisions on configuration selection and system performance and to help identify those system elements whose high costs merit special management and design attention. LCC analyses are a continuing activity for monitoring and refining original estimates, for evaluating proposed and imposed design changes, and for augmenting technical trade studies at finer levels of detail.

Cost models are generally used to make LCC estimates. The three model types are forecast sensitivity or parametric, cost estimating relationships (CER), and accounting. A forecast sensitivity model uses equations developed from analysis of past experience (e.g., RCA "PRICE" is such a model). It can predict cost sensitivity to changes in schedule, parts quality, etc. A CER is an equation that predicts cost in terms of performance or design parameters (e.g., MTBF, weight, data rate). An accounting model is a set of equations that express the cost in terms of the estimated actual labor or materials expended (e.g., design cost as a function of the various man-hours budgeted for circuit design, breadboard fabrication, laboratory test, etc.). LCC estimates are usually performed by LCC personnel, who are most cognizant of applicable models. These specialists, who respond to program office and contract requirements, must be thoroughly briefed on the multiplex system characteristics in order to make intelligent choices of models. LCC analysis is therefore a team effort of LCC model specialists and project engineers. The system engineer must inform the analyist of the design trade-offs which must be considered, so that the LCC personnel can evaluate potential models' capability to support these trades. Inputs to the LCC models are supplied by engineering, finance, and logistic support organizations depending on the cost model requirements. Complex cost models may be computerized, so programming support may also be required.

Appendix D describes cost elements in the life cycle, discusses cost models in terms of both sources and several specific models that may be helpful, presents a reliability prediction model for airborne multiplex systems, and discusses reliability cost-optimization.

3.3 MULTIPLEX SYSTEM REQUIREMENTS AND DESIGN SPECIFICATION

Once the multiplex design decisions have been made, they must be documented as part of the system specifications. A hardware specification, a software specification, and an interface specification must be documented. The requirements for hardware and software specification documentation are well defined in MIL-STD-483 and MIL-STD-490. The interface specification, however, is a document with content that is oriented specifically to the bus communication. Each bus communication must be documented at the message level, the word level, and the bit level. Examples of this documentation are shown in figures 3.3-1, 3.3-2, and 3.3-3. Figure 3.3-1 describes the command word formats and assigns the entire set of subaddresses used for the

COMMAND WORD

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

BIT TIME

SYNC (bits 1–3) · TERMINAL ADDRESS (bits 4–8) · SUBADDRESS MODE (bits 10–14) · DATA WORD COUNT (bits 15–19) · PARITY BIT (bit 20)

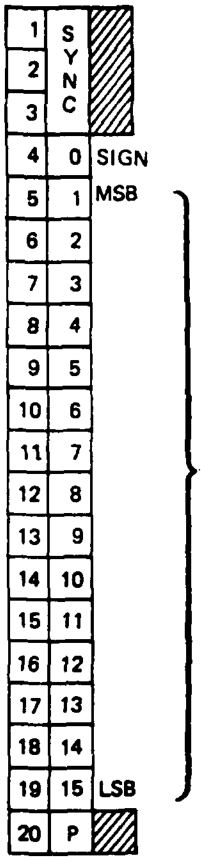
TRANSMIT/RECEIVE
"1" = TRANSMIT
"0" = RECEIVE

ALWAYS "1" FOR INSTRUMENTED AIRCRAFT
(Used by instrumentation and data processor to indicate WORD is: "1" = COMMAND and "0" = STATUS)

T/R	Always "1"	Subaddress				BLOCK	SUBSYSTEM
0	1	0	0	0	0	F03	FIRE CONTROL RADAR
0	1	0	0	0	1	I05	
0	1	0	0	1	0	S04	
0	1	0	0	1	1	Z01	
1	1	0	0	0	0	R01	
1	1	0	0	0	1	R02	
1	1	0	1	0	0	R03	
1	1	0	0	1	0	R04	
1	1	0	0	1	1	R05	
0	1	1	1	0	1	C02	HEAD-UP DISPLAY
0	1	1	1	1	0	F08	
0	1	1	0	0	0	F09	
0	1	1	0	0	1	F10	
0	1	1	0	1	0	I04	
0	1	1	0	1	1	R03	
0	1	1	1	0	0	S02	
1	1	0	0	0	1	H01	
0	1	0	0	0	0	F07	STORES MANAGEMENT SUBSYSTEM
0	1	0	0	0	1	R02	
1	1	0	0	0	0	S01	
1	1	0	0	0	1	S02	
1	1	0	0	1	0	S03	
1	1	0	1	0	0	S04	
0	1	0	1	0	1	F05	RADAR/EO DISPLAY SET
0	1	0	1	1	0	F13	
0	1	0	1	1	1	F15	
0	1	0	1	0	0	R04	
0	1	0	0	1	1	S03	
1	1	0	0	0	1	Z01	
1	1	0	0	1	0	Z02	
0	1	0	0	1	1	C03	INERTIAL NAVIGATION UNIT
0	1	0	1	0	0	F01	
0	1	0	0	0	1	F02	
0	1	0	1	1	0	F12	
0	1	0	1	0	1	F16	
0	1	0	0	1	0	F17	
0	1	0	0	0	0	P02	
1	1	0	0	0	0	I01	
1	1	0	0	0	1	I02	
1	1	0	0	1	0	I03	
1	1	0	0	1	1	I04	
1	1	0	1	0	0	I05	
1	1	0	0	0	0	F04	FIRE CONTROL AND NAVIGATION PANEL
1	1	0	0	0	1	I03	
1	1	0	0	1	0	P01	
1	1	0	0	1	1	P02	
1	1	0	0	0	0	C01	CENTRAL AIR DATA COMPUTER
1	1	0	0	0	1	C02	
1	1	0	0	1	0	C03	
1	1	0	0	0	0	F14	TARGET IDENTIFICATION SET, LASER
1	1	0	0	0	1	E01	

Figure 3.3-1. Command Word/Subaddress Mode

1553 DATA WORD



Designation: MACR antenna sine roll 40100202
Command word designation: NAWD-DEU data
Word 2 of 7 words

Format: 2's complement, fractional
Units: sine of angle (unit)
Maximum value: +0.99996949
Minimum value: -1.0
Accuracy: 2^{-15} (value of LSB)
MSB (positive) = 0.5

Orientation:

0.0 corresponds with "wings level"
+0.5 corresponds with 90-deg right bank
-0.5 corresponds with 90-deg left bank

Figure 3.3-2. 1553 Data Word Description Example (B-1)

Radar Set Control Word

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01		-	RCVRGAIN			-		RDRMODE						SI	-

BITS	FIELD	DESCRIPTION	UNITS	SCALING
15-14		Word Identifier = 1		
13		Spare		
12-10	RCVRGAIN	Receiver Gain 1 = Level 1 2 = Level 2 3 = Level 3 4 = Level 4 5 = Level 5 6 = Level 6 7 = Level 7		
9-8		Spare		
7-5	RDRMODE	Radar Mode 001 = STBY 101 = Mode I 100 = Mode II 111 = Mode III		
4-2		Spare		
1	SI	Signal Integration 0 = On 1 = Off		
0		Spare		
REMARKS				

AN/AYK-14 to SDC

Figure 3.3-3. Data Word Description Example

subsystems. Figure 3.3-2 describes word 2 (as an example) of a 7-word message. Note that a concise definition of that data word is given in the format, units, maximum value, minimum value, accuracy, and orientation. An alternative format for the definition of a data word is specified in figure 3.3-3. In this example, several fields are defined within a particular word of a particular message.

Another specification oriented specifically toward the data bus is the Data Bus Interface Requirements. This document, sometimes called the protocol document, is based on the MIL-STD-1553 standard and makes more concrete the bus controller and remote terminal definitions. Addresses for each device, mode codes that will be used, and requirements for time tagging of data (e.g., minor cycle numbers) are defined. In 1553A, the status field was undefined; in 1553B the optional status bits require no further definition, but are application dependent.

3.4 MULTIPLEX SYSTEM TEST

Although testing a multiplex system is not normally included in the activities associated with system design, except for the preparation of test requirements, the subject is important enough to present some necessary concepts and current practices. There are some avionics integration design activities that are supported by laboratory tests (e.g., bus network modeling). Tests of 1553 aircraft data bus system will take place (1) at the facilities of suppliers of LRU's with 1553 interfaces, (2) at a "hot bench" generally located at the airplane company, and (3) during flight test. Flight tests will usually be conducted near the airplane company and possibly at military bases and other locations by the eventual military using command. The purpose of this section is to briefly describe hardware, test procedures, and test philosophy of the various levels of testing that have been found useful for tests of 1553 data bus systems.

3.4.1 Scope of Tes s

Data bus interfaces are not usually designed, developed, and tested independently of all other LRU interfaces. Although the discussion centers on tests of subsystems with 1553 interfaces and system test, it should be understood that many other design and test activities are required to successfully complete avionics integration for an airplane. The 1553 terminal design and test form a part of these activities. This is so because 1553 defines a terminal as: "The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus." Tests of subsystems connected to the 1553 data bus usually include verification of interface functions on the subsystem side as well as the bus side of the terminal.

Design verification tests of 1553 terminals are an important part of system and subsystem design. Table 3.4-1 is a summary of design procedures. Table 3.4-2 is a summary of test procedure development. Each table is divided into two parts, procedures for the subsystem side as well as the bus side of the terminal. The procedures presented in the tables should be used to scope the test activities.

Table 3.4-1. Design Procedures Guidelines

1. Design procedures for avionic subsystem interface to remote terminal (RT):
   a. The Government or the integration contractor will provide the standard interface modules' (IM) definition to the avionic contractor, when standard IM's are used.
   b. Contractor will develop the electrical interface definition between the avionic subsystem and the RT IM.
   c. Perform an engineering design of the electrical interface.
   d. Manufacture the breadboard to accommodate a standard IM.
   e. Develop support test equipment to generate the IM interface input and output signals.
   f. Define acceptance test.
   g. Perform acceptance test.
   h. Evaluate test results.
   i. Report out-of-limits conditions for failed test.
   j. Apply test results to reevaluate and improve interface performance.

2. Design procedures for avionic subsystem interface to MIL-STD-1553 bus:
   a. Evaluate interface definition of chosen avionic subsystem.
   b. Design and develop the electrical interface definition between the avionic subsystem and the 1553 bus.
   c. Design a bus interface unit and subsystem interface.
   d. Develop support test equipment to generate the subsystem and bus input and output signals.
   e. Define acceptance test.
   f. Perform acceptance test.
   g. Evaluate test results.
   h. Report out-of-limits conditions for failed test.
   i. Apply test results to reevaluate and redesign interface performance.

3.4.2 Typical 1553 Bus Checkout Systems

The purpose of this section is to introduce the types of bus checkout systems that are frequently used. Generally, there are three classes:

a. The bench or suitcase 1553 multiplex bus tester
b. The entire avionics hot bench
c. The programmed bus monitor for flight test

The bench or suitcase testers may also be used to support troubleshooting during hot bench and flight tests.

3.4.2.1 Multiplex Bus Tester and Simulator

A simulator is a versatile data bus test instrument compatible with MIL-STD-1553 for applications in the engineering laboratory, in system integration laboratories, and as a portable instrument for fault isolation.

Simulators have full capability to act as a bus controller, both sending and receiving data bus messages. The simulator will transmit a command word and a selected number of 16-bit data words. The command word is front panel selectable, and the 16-bit field of each data word is loaded into memory from the front panel switches. The proper polarity sync patterns and parity bits are added to each word to provide the correct word formats.

Table 3.4-2. Test Procedures Guidelines

1. Methods for developing test procedure for remote terminal (RT) to avionic subsystem interface:
    a. Coordinate with interface equipment designers.
    b. Identify subsystem interface requirements.
    c. Determine nominal interface operation.
    d. Identify error modes and off-nominal operations.
    e. Determine desired form of test results.
    f. Determine test equipment requirements.
    g. Establish system time lines within protocol constraints.
    h. Flow chart test requirements.
    i. Recommend test procedures.

2. Method for developing test procedure for a MIL-STD-1553 to avionic subsystem interface:
    a. Coordinate with interface equipment designers.
    b. Identify subsystem interface requirements.
    c. Determine nominal interface requirements.
    d. Identify error modes and off-nominal operation.
    e. Determine desired form of test results.
    f. Determine test equipment requirements.
    g. Establish system time lines within MIL-STD-1553 protocol constraints.
    h. Determine MIL-STD-1553 parameters that need testing.
    i. Recommend test procedures.

Simulators also have the features necessary to receive the message from a remote terminal, corresponding to the command word address that was transmitted. The unit will display the status and all data words as selected from the front panel control switches.

Simulators have diagnostic and off-nominal capability. Typically, the following invalid messages and words can be transmitted to another terminal to determine its response:

a. Invalid command word
b. Invalid data word
c. Invalid word count
d. Invalid number of bits (usually 15 or 17)
e. Invalid parity
f. Nonvalid manchester encoding

Simulators have the capability to receive and verify valid and inv[illegible] words and messages.

3.4.2.2 Avionics "Hot Bench"

Hot bench is a commonly used term to descri[illegible] a [illegible] laboratory (SDL) or system integration laboratory (S[illegible] SDL's provide simulation capability and data recor[illegible] used in the development of avionic hardware and [illegible] incremental testing of avionics interfaces, [illegible]

subsystems such as the radar or stores are used. The simulators provide realistic inputs and responses so that dynamic conditions may be evaluated in the laboratory. This is in contrast to the capability of bench or suitcase testers, which usually can evaluate only a command and a response. The simulators in hot benches are a substitute for unavailable hardware, and an intermixing of prototype or production airplane hardware with simulators is usually possible. By this means, airplane hardware can be incrementally added to an avionic system. Whenever the interface is the 1553 data bus, rapid resubstitution of the simulator for the airplane hardware permits the isolation of problems or anomalies.

Several benefits of integration with 1553 data buses become apparent during system test. The data bus approach requires integrating only one electrical interface per subsystem versus multiple interfaces in the point-to-point method. The single interface also allows more of the integration activity to be done at the subsystem level using one special test fixture, which might be too costly with many unique point-to-point interfaces. Simulating the data bus interface of subsystems can easily be done using a computer with a data bus interface as the simulator. The equivalent simulation in a point-to-point architecture may require several special-purpose interfaces to be developed. The data bus will also accommodate unexpected integration problems such as added data words, changes in update rates, and rerouting of data parameters.

3.4.2.3 Bus Monitor and Airborne Instrumentation

System designers should make provision for the connection of a bus monitor and avionics instrumentation capability. Provision will usually be a stub or connection properly terminated when not in use on prototype and test airplanes. With this connection available, a bus monitor may be used during flight test to acquire selected bus messages. Recall that 1553B, paragraph 4.4.4, describes bus monitor operation.

> 4.4.4 Bus Monitor Operation
>
> A terminal operating as a bus monitor shall receive bus traffic and extract selected information. While operating as a bus monitor, the terminal shall not respond to any message except one containing its own unique address if one is assigned. All information obtained while acting as a bus monitor shall be strictly used for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller.

Bus monitors are usually implemented using a digital computer, appropriate memory for buffering, and magnetic tape for recording. Several suppliers of bus monitors and airborne instrumentation are qualified for flight test.

3.5 ROADMAP TO MULTIPLEX SYSTEM DESIGN

The purpose of the roadmap is to relate the multiplex system design activities to a normal defense system acquisition. Multiplex system design as related to conceptual, validation, and full-scale development phases of a normal system acquisition is discussed.

The aircraft avionics of modern military airplanes must be viewed as an integrated system rather than a conglomerate of functional sensors, controls, and displays. The data bus integration method forces the system engineer to perform system-level analysis, because information flow definition is the key element in the orderly integration process. When using the data bus concept, it is important to realize that a large amount of the integration is accomplished through software. Therefore, the final and most critical integration step is performed through the effective application of flight software (e.g., the software controls the real-time information flow). Therefore, to avoid schedule delays and cost overruns that may be associated with insufficient planning for software and computer systems, it is important that the critical technical decisions relating to data bus system integration be made in the phase that supports this intelligent planning. If the military data bus standard (MIL-STD-1553) is to be effective, its definition in the Program Management Directive and the Program Management Plan is essential. The system acquisition life cycle provides a basis for categorizing program management activities. It consists of five major phases with major decision points as defined in AFR 800-2. Using this as a basic management approach, the following descriptions briefly explain a normal system acquisition with emphasis on multiplexing and computer resources. This discussion is adapted from AFR 800-14, Management of Computer Resources in Systems.

The conceptual phase is the initial planning period when the technical, military, and economic bases are established through comprehensive studies, experimental development, and concept evaluation. The objective of this initial planning may be directed toward refining proposed solutions or developing alternative concepts to satisfy a required operational capability.

During this phase, proposed solutions are refined or alternative concepts are developed using feasibility assessments, estimates (cost and schedule, intelligence, logistics, etc.), trade-offs, studies, and analyses.

The major definitive document resulting from this phase is the initial system specification, which documents total system performance requirements. It will document the requirements for integration (e.g., avionic sensors, crew displays, weapon delivery systems, etc.). The initial system specification will be used to establish the general nature of the system that is to be further defined during a contract definition activity. This specification will be the basis for the performance required of prime items and configuration items, since overall system performance will be allocated to these items. It is to be expected that the following definition of applicable requirements related to a data bus system would be included in the initial system specification:

a. A requirement stating that MIL-STD-1553 data buses will be used in the systems integration of aircraft subsystems

b. Requirements for subsystems to be connected to the data buses, such as inertial navigation systems, fire control systems, crew displays, computers, radio control heads

c. System-level redundancy requirements

d. A description of the multiplex topology, including line drawings

e. A description of the overall sensor and data bus system control approach

The system specifiction may also document the requirements to be met by computer resources as well as relevant design constraints. An adequate definition of essential system interfaces between computer equipment functions, communication functions, and personnel functions should be provided to enable the further definition and management of the computer programs and computer equipment into configuration items. Normally, this information is derived from system engineering studies of the system functions.

The validation phase is the period when major system characteristics are refined through studies, system engineering, preliminary equipment, computer program development, test, and evaluation. The objective is to validate the choice of alternatives and to provide the basis for determining whether or not to proceed into the next phase.

During this period, system performance requirements including computer resources are further defined, and preferred development methodologies for computer programs, such as organic or contractor (per AFR 26-12), are selected. Validation phase activities define the efforts required by characteristics (performance, cost, and schedule) and provide confidence that risks have been resolved or minimized. Technical reviews that should be accomplished are the system requirements review(s) and system design review.

For computer resources, the major definitive documents resulting from this phase are the authenticated system specification, the preliminary development specifications containing system functional requirements allocated to configuration items of computer programs and equipment, and the initial computer resources integrated support plan (CRISP).

For the data bus system, the major definitive documents resulting from this phase are the same as for computer resources. The system specification should contain additional detail on--

a. Whether the data bus interface unit is standalone or part of the sensor unit

b. Requirements for growth of bus control program sizing and throughput

c. Overall definition of all normal and error recovery data communications, identifying at least the source, destination, update rate, and nominal length of each message

d. Requirements for growth of the data bus system

e. Requirements for addition of remote terminals

f. Definition of the applicable portions of MIL-STD-1553B that are optional

g. Rationale for deviations from MIL-STD-1553B

h. Requirements for the use of existing subsystems (whether GFE or not)

The full-scale development phase is the period when the system, equipment, computer programs, facilities, personnel subsystems, training, and the principal items necessary for support are designed, fabricated, tested, and evaluated. The intended outputs are a system that closely approximates the production item, the documentation necessary to enter the production phase, and the test results that demonstrate that the system to be produced meets the stated performance requirements.

The development specifications are completed and authenticated. Authentication of any development specification establishes the allocated baseline. A preliminary design effort is accomplished leading to an acceptable design approach. A preliminary design review (PDR) is held for each equipment configuration item and computer program configuration item (CPCI) to review the preliminary design against the respective authenticated development specification. Formal engineering change control procedures are implemented to prepare, propose, review, approve, implement, and record engineering changes to the allocated baseline. For computer programs, the preliminary design includes the definition of the entire computer program in terms of functions, external and internal interfaces, storage allocation, computer program operating sequences, and the design of the data base. This information should be contained in the development specifications and become the basis for the PDR of the computer program. For data bus systems, preliminary design includes a complete date transfer description usually in the form of an interface control document, a definition of the design approach of each remote terminal, and a definition of the bus control software and its functions in relation to the system executive software. A significant item to evaluate at PDR is the proposed methods and mechanizations of backup and alternative bus control switchover.

Following an acceptable PDR for a configuration item, detailed design of that item begins. This activity produces engineering documentation such as drawings, product specifications, and test plans. For computer programs, design is accompanied by documentation of logical flows, functional sequences and relations, formats, constraints, and the data base. This documentation should be reviewed by engineering personnel prior to the critical design review (CDR). The CDR should ensure that the recommended design satisfies the requirements for the development specification. The primary product of the CDR for CPCI's is the identification of specific portions of the product specification, which will be released for coding and testing. For data bus systems, the CDR should ensure that each assigned data transfer can be accomplished in the proposed design and that all requirements for message growth and system growth have been consistently applied in the design.

Development, test, and evaluation (DT&E) and initial operational test and evaluation (IOT&E) are conducted. Testing of configuration items is performed according to formal test plans initially submitted in preliminary draft form for review at CDR and finalized prior to the start of testing. These activities normally proceed in such a way that testing of selected functions begins early during development and proceeds through successively detailed levels of assembly to the point where the complete system is subjected to formal qualification testing. A data bus system allows considerable flexibility in system testing, even to the point that the same facility may be used alternately for subsystem qualification, system integration, and software qualification. Configuration management of the

test facility and the software is vital to integration test. Each additional terminal (remote terminal or sensor) that is added requires verification of the simulation, usually to be done simultaneously with initial subsystem test. Additional computer programs and equipment may be required to properly simulate the operational environment or test the computer programs or sensors. The scope and realism of computer program testing may be progressively expanded as additional items of the operational computer equipment and subsystems are made available for this purpose. Adequacy of the performance of these systems is checked to the maximum extent possible through prudent use of simulation prior to the installation of the sensor into the integration. The use of artificial data or recorded data from similar equipment should be considered. Nuclear safety cross-check analysis (NSCCA) is also performed on specified computer resource items (AFR 122-1). Satisfactory performance of the system, especially a large operational system, may not be completely demonstrated and assessed until completion of operational test and evaluation (OT&E).

Planning for transfer of the system to the supporting command and turnover to the using command begins early in this period. Necessary agreements should be prepared, coordinated, and approved prior to the end of this phase.

CHAPTER 4
HARDWARE DESIGN

Table of Contents

Table of Contents (Continued)

List of Figures

List of Tables

4.0 HARDWARE DESIGN

4.1 Multiplex Terminal Definition and Functional Partitioning

As an introduction to the hardware design section, the various multiplex data bus terminal units are defined and a generalized terminal is described showing the partitioning of the major functional elements. The description of remote terminals, bus controllers, and bus monitors related to typical system operation will provide the basis for the detailed hardware implementation discussed in the paragraphs to follow.

4.1.1 Definitions

Since the multiplex data bus was conceived, a great deal of confusion has existed relative to the terminology for components of the system. This is especially true of the definitions for units that interface the bus to user subsystems. MIL-STD-1553A defines two types of interface units, remote terminals (RT) and controllers, with the implication that the RT is any subsystem bus interface not performing the function of the bus controller. An attempt has been made in MIL-STD-1553B to provide clearer and more complete definitions for the various types of interface units in a typical data bus system.

Paragraph 3.10 of MIL-STD-1553B defines a terminal as a unit that provides an interface between the data bus and a subsystem. The types of terminals -- (1) RT, (2) bus controller, and (3) bus monitor -- are described on the basis of their function in the overall system operation, not in relation to physical characteristics or location. The remote terminal is defined as any terminal not operating as the bus controller or as a bus monitor. The RT functions in response to a command word received from the bus controller if the terminal address corresponds to the predetermined unique RT address. The RT may receive or transmit data words and responds with a status word for all commands except for broadcast. The bus controller is defined as the terminal operating as the initiator of information transfers (messages) on the data bus. A bus monitor is defined to be a terminal that receives bus traffic and extracts selected information to be used at a later time. It is possible that a physically identifiable flexible terminal unit may be programmed to function as an RT, bus controller, or monitor during the course of system operation, or the terminal unit may be designed to perform only one or two of the three functions. Therefore, the designation of a terminal type may change for interface units that perform difficult functions for different system operating modes. As an example, a "smart" subsystem may serve as an RT or bus monitor during normal system operation and may take over system operation, acting as the bus controller, when the primary bus controller fails. In other applications, such as distributed processing systems, various subsystems may provide control of the bus. This mode of operation is also defined as "dynamic bus control" in MIL-STD-1553B.

4.1.2 Functional Elements and Interfaces

From the previous discussion, it is apparent that a terminal may be specifically designed as an RT, bus controller, or monitor unit. A more flexible terminal design is also possible, which will allow it to perform any of the three roles (i.e., RT, bus controller, or monitor). The

following paragraphs describe the functional elements and interfaces of a generalized terminal design.

Figure 4.1-1 identifies the major functions of the generalized terminal. Functions are represented rather than the architecture, which is independent of implementation.

Four major functional elements are defined for the generalized terminal: (1) analog transmit and receive (A), (2) bit and word processor (B), (3) word and message processor (C), and (4) subsystem interface circuits (D). The transfer of information between the functional elements may be in serial or parallel form. It is certainly possible that some of the word and message processing and subsystem interface functions can be integrated into certain types of subsystems such as computers or smart processors. The requirements for the subsystem interfaces are different for a digital subsystem than for a subsystem with distributed and widely varying types of signal sources and loads such as an electrical multiplex (EMUX) or test system.

4.1.2.1 Analog Transmit-Receive

The analog transmit-receive functional element (A) is primarily the analog front end required to interface digital logic with the data bus. Even though the transmitted signal on the bus is generated in digital form, the twisted-shielded pair transmission line characteristics, along with multiple terminations, result in a signal received by the terminal that is attenuated and typically approaches a distorted sine wave. This element contains the coupler components required for connection to the bus, fault isolation resistors, and coupling transformer as required by MIL-STD-1553. The input signal from the bus is filtered to remove any high-frequency noise that may have been induced on the signal in the bus network. The threshold detector allows for low-level noise rejection and provides a digital output compatible with the detection logic that follows. The transmitter driver is furnished with a control signal and biphase-modulated signals in the data and status word format defined by MIL-STD-1553. A timer is provided to shut down the transmitter driver after a predetermined period to prevent a condition of uncontrolled "chattering" on the data bus. The transmit-receive element of a data bus terminal is implemented with discrete or hybrid circuits because of the type of components required (i.e., transformers, resistors, capacitors, and power transistors).

4.1.2.2 Bit and Word Processor

The bit and word processor element (B) functions indicated are required to process the bits and words of the information flow from and to the data bus. The squared up signal from the receiver threshold detector is input to the element (B), which is sometimes referred to as the encoder-decoder. The decoder portion senses bit timing to decode the word synchronization pattern, including polarity, to identify command words and data words. Bits other than sync are checked to verify that the signal is in compliance with the Manchester (biphase) coding. The number of bits per word and parity are checked. Error conditions are flagged if these data validation checks indicate an out-of-tolerance condition. The encoder or transmit portion of the bit and word processor element contains the logic for status and data word formatting, including sync and parity generation. Control signals are

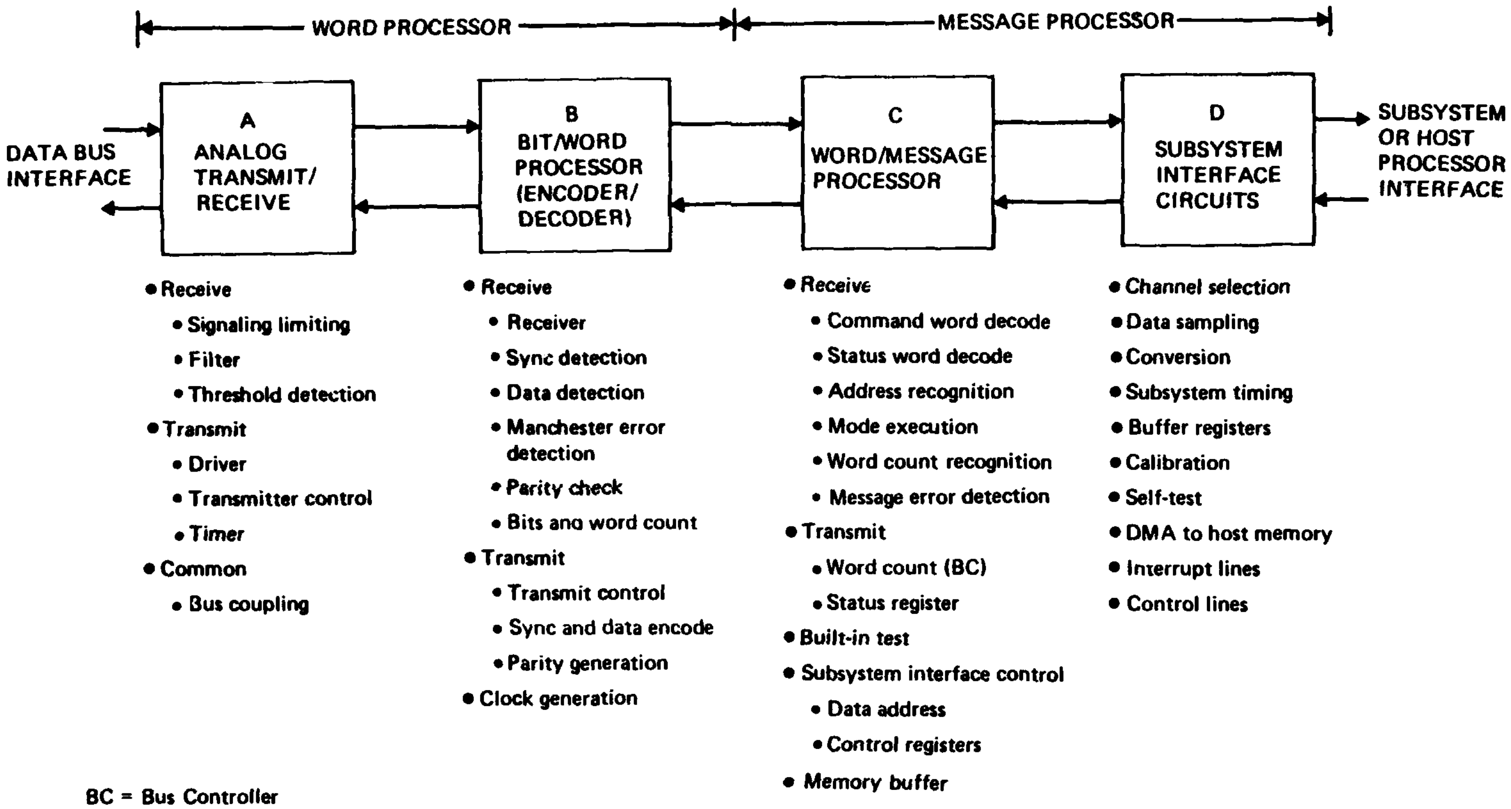


Figure 4.1-1. Generalized Terminal Functional Elements

also provided for the transmitter in the transmit-receive element (B). A crystal-controlled clock generator is usually included as a source for terminal timing.

The bit and word processor (encoder-decoder) has been implemented in a variety of ways. Early designs of the decoder employed analog techniques for the bit timing integration, which resulted in temperature and aging drift problems and unwieldy component packaging. More recent designs employ digital techniques that have led to large-scale integration (LSI) implementation of the encoder-decoder functions. A number of commercially available LSI devices have been introduced.

4.1.2.3 Word Processor Unit

The combination of the analog transmit-receive element and the bit and word processor element provides a bus interface function that is adequate for incorporation in some smart subsystems and is sometimes referred to as the "front-end" word processing terminal. This portion of a terminal is transparent to content of the word and provides the reconstructed words with error indications for further processing by the word and message processor. For purposes of this discussion, the division of functions has been chosen so that word processing functions are independent of bit assignments, field assignments, sync polarity, sequence of words, and message or word gaps. They are dependent on word length, location of word sync, and sync and bit encoding. A number of examples of this type of terminal module developed for programs or as standard products can be cited. The space shuttle employs a common design for interfacing a wide variety of subsystems and is termed the multiplex interface adapter (MIA). Standard product units that perform the word processing function have been developed and produced. One of these modules is referred to as a multiplex terminal interface (MTI). Another large company has designed an LSI front-end module for a variety of in-house applications with the added capability of command word address recognition. The avionics multiplex interface for the B-1 bomber program uses a similar design.

4.1.2.4 Word and Message Processor Subsystem Interface

Command, data, and status words may be transferred between the bit and word processor and the word and message processor functions in serial or parallel form. If the word processor front-end module is to be used with a variety of digital subsystems, the parallel format is usually preferred. Certain units, especially the standard products, have been furnished with both serial and parallel interface, selectable by the user.

Some of the functions assigned to the word and message processor element (C) (such as command word decoding and address recognition) can be performed by the bit and word processing element as indicated above. The functions assigned to the word and message processor and subsystem interface circuits elements are oriented to message processing. The information content of the words associated with the overall message is decoded and used for control of information flow between the terminal and subsystem. The command word is decoded to determine if the terminal has been addressed. If the terminal is operating as an RT and the received address does not match the preprogrammed terminal address code, no further message processing is required. The terminal that is serving a subsystem operating as a bus controller is

programmed to process status words and associated messages. A monitor terminal will process the message for those command word terminal addresses that correspond to one of a preprogrammed set of address codes. Further decoding of the command word yields (1) RT transmit or receive instructions, (2) RT subaddress or optional mode control designation, and (3) message data word count or mode control code.

If the message data word count is contained in the command word, the number of words received from the bus controller is checked for input to the message error indication. The status register and control logic is included in this element. The message error bit (bit 9) of the status register is set to "one" for conditions of word errors, transmission discontinuity, illegal command (optional for RT), and invalid data reception as required by MIL-STD-1553B. The status word is formatted and transmitted to the word processor and subsequently to the bus controller upon reception of a valid command word and contiguous data words meeting the validity criteria. In some cases it may be required to verify a complete message prior to data transfer between the terminal and subsystem. A message buffer memory of 32 words is provided in the word and message processor for message validation. Word count generation is required for a terminal issuing the command word when serving as the bus controller interface.

The built-in-test capability of terminals will vary, depending on their application to subsystems and the approach to hardware implementation. The status response word contains, in addition to message error indication, a subsystem flag bit and a terminal flag bit that indicate a subsystem and terminal fault condition, respectively. In addition, certain test conditions can be implemented to perform end-to-end performance checks. An effective test that is often implemented involves generation of a test message in the subsystem or message processor, transmission of the message to the bus interface, reception of the test message, and comparison of the received message at the point of origin. A microprocessor-controlled message processor offers a significant capability for efficient and effective built-in-test implementation. A major task of the word and message processor function is the subsystem interface control. The subsystem interface complexity can vary from a simple parallel "handshake" for communication using a digital computer I/O to a wide variety of signal conditioners to interface with analog, discrete, frequency and asynchronous serial digital sources and loads. These latter signal requirements are typical of existing avionic subsystem interfaces and aircraft test parameters. For example, the terminal or subsystem RT required to service an airborne radar system (avionic subsystem) could be a simple serial or parallel digital interface; such an RT would be a plug-in module contained in the radar subsystem electronics housing. An RT required to service an airframe or electrical subsystem with various types of sensors and loads spread over a geographical area is required to be a standalone unit with internal power supply. A variety of multiplexers, demultiplexers, A/D converters or D/A converters might be required to interface such an RT with the subsystem signal sources and loads.

More detailed description of terminal types, including examples of RT, bus controllers, and bus monitors that have been implemented for advanced aircraft systems, is included in section 4.5.

4.2 BUS NETWORK AND TERMINAL INTERFACE CONSIDERATIONS

One of the most significant considerations facing the data bus system designer and integrator is the definition of the bus network and specification of the terminal interfaces. The bus network must be designed for signal integrity to achieve a bit error and word error rate performance within that specified and to provide fault isolation and redundancy for the required system reliability. The characteristics of the terminal interface must be specified for assurance of acceptable system performance when terminals are connected to the network in all possible operational configurations. The following discussion provides a summary and comparison of MIL-STD-1553A/B requirements that have significant effect on the terminal hardware design, especially at the bus interface. The effect of stubbing from the main bus to the terminal is reviewed to show why external coupler hardware may be required to maintain signal integrity and fault isolation characteristics of the bus network.

A discussion of shielding and grounding techniques for electromagnetic interference (EMI) control is presented to furnish guidance for specification.

4.2.1 MIL-STD-1553A/B Bus Network and Terminal Interface Requirements

Some confusion relative to data bus network and terminal interface requirements has resulted from requirements contained in MIL-STD-1553(USAF) and MIL-STD-1553A. The rationale will be presented for additions and changes to the data bus network and terminal interface characteristics incorporated in 1553B. The B-version is significantly influenced by "lessons learned" experience gained in the implementation of systems using the original and A-revision of the standard.

4.2.1.1 Data Bus Network

Table 4.2-1 is a summary listing of the data bus and coupling requirements contained in 1553A and 1553B revisions. The characteristics of the twisted-shielded pair cable have been relaxed to allow selection of cable types from a variety of manufacturers. It has been shown that minor variations from the specified cable characteristics do not significantly affect the system performance.

A great deal of concern is attributed to the cable network requirements, including bus length, coupling, and stubbing. The original version of the standard and the A-revision did not provide sufficient guidelines for bus network design, especially for the transformer coupled stub. A maximum cable length of 300 ft was assigned for the main bus. The B-version does not specify a maximum main bus length because the cable length, number of terminals, and length of stubs are all subject to trade-off and must be considered in the design for reliable system operation. In other words, an arbitrary limit of 300 ft should not be applied because all parameters of the network must be considered.

A generalized multiplex bus network configuration is shown in figure 4.2-1. The main bus is terminated at each end in the cable characteristic impedance to minimize reflections because of transmission line mismatch. With no stubs attached, the main bus looks like an infinite length transmission

Table 4.2-1. Summary of Data Bus and Coupling Requirements

Parameter	MIL-STD-1553A	MIL-STD-1553B
• Transmission line		
• Cable type	Twisted-shielded pair	Twisted-shielded pair
• Capacitance (wire to wire)	30 pF/ft, maximum	30 pF/ft, maximum
• Twist	One per inch, minimum	Four per foot (0.33/in), minimum
• Characteristic impedance (Z_o)	70 ohms ±10% at 1.0 MHz	70 to 85 ohms at 1.0 MHz
• Attenuation	1.0 dB/100 ft at 1.0 MHz, maximum	1.5 dB/100 ft at 1.0 MHz, maximum
• Length of main bus	300 ft, maximum	Not specified
• Termination	Two ends terminated in resistors equal to Z_o	Two ends terminated in resistors equal to Z_o ± 2%
• Shielding	80% coverage, minimum	75% coverage, minimum
• Cable coupling		
• Stub definition	Short stub <1 ft Long stub >1 to 20 ft (20 ft maximum)	Short stub <1 ft Long stub >1 to 20 feet (may be exceeded)
• Coupler requirement	All connections use external coupler box; connectors specified (ref. fig. 4.2-1)	Direct coupled—short stub; transformer coupled—long stub (ref. fig. 4.2-2)
• Coupler transformer		
• *Turns ratio*	*Not specified*	1:1.41
• Input impedance	Not specified	3,000 ohms, minimum (75 kHz to 1.0 MHz)
• Droop	Not specified	20% maximum (250 kHz)
• Overshoot and ringing	Not specified	±1.0V peak (250-kHz square wave with 100-ns maximum rise and fall time)
• Common mode rejection	Not specified	45.0 dB at 1.0 MHz
• Fault protection	Resistor in series with each connection equal to ($0.75Z_o$) ± 5% ohms	Resistor in series with each connection equal to ($0.75\ Z_o$) ± 2.0% ohms
• Stub voltage	±0.5V to ±10.0V, peak l-l (1.0V to 20.0V, p-p, l-l); nominal signal level for terminal response	1.0V to 14.0V p-p, l-l, minimum signal voltage (transformer coupled); 1.4V to 20.0V, p-p, l-l, minimum signal voltage (direct coupled)

line, and there are no disturbing reflections. When the stubs are added for connection of the terminals, the bus is loaded locally and a mismatch occurs with resulting reflections. The degree of mismatch and signal distortions because of reflections is a function of the absolute impedance (Z) presented by the stub and terminal input impedance. To minimize signal distortion, it is desirable to maintain a high stub impedance reflected back to the main bus. At the same time, the impedance needs to be kept low so that adequate signal power will be delivered to the receiver input. A trade-off and compromise between these conflicting requirements are necessary to achieve the specified signal-to-noise ratio and system error rate performance. Two methods for coupling a terminal to the main bus are defined in 1553B, transformer coupling and direct coupling (see fig. 4.2-2). Transformer coupling is usually used with long stubs (1 to 20 ft) and requires a coupler box, separate from the terminal, located near the junction of the main bus and stub. Direct coupling is usually limited for use with stubs of less than 1 ft. Fault isolation resistors (R) are included to provide protection for the main bus in case of a short circuit in the stub or terminal. The coupler-transformer characteristics defined in 1553B and listed in table 4.2-1 provide a compromise for the signal level and distortion characteristics delivered to the terminals. The coupler-transformer turns ratio (1:1.41) provides beneficial impedance transformation for both terminal reception and transmission.

A plot of the calculated first-order-magnitude stub absolute impedance (Z) versus stub length is presented in figure 4.2-3. As indicated, the improvement of stub load impedance is a result of impedance transformation that is proportional to the square of the turns ratio, assuming an ideal coupler transformer. The band of curves for the transformer-coupled case indicated by the darkened area between the curves results from assuming various values of transformer shunt impedance. The lower bound is the curve using a transformer with the minimum impedance (3,000 ohms) specified in 1553B. The upper bound is for an ideal transformer with very high impedance. All values of stub impedance are magnitude values for a 70-ohm cable with 30 pF/ft capacitance and are calculated for 1,000 ohms terminal input impedance, with the exception of the upper direct-coupled curve. This curve is based on the 1553B specified terminal input impedance of 2,000 ohms. It can be seen from these curves that stub impedance values are increased generally by use of the coupler transformer which provides at least a 2 to 1 improvement for the longer (> 10 ft) stubs. The curves also show the importance of the transformer characteristics for maintaining the expected improvement.

As indicated above, the 1:1.41 transformer also provides ideal termination of the stub for transmission of signals from the terminal to the main bus. Impedance at the main bus is

$$Z_B = \frac{Z_o}{2} + 2R \qquad (1)$$

where,

$$R = 0.75\ Z_o$$

$$Z_B = 0.5Z_o + 1.5Z_o = 2Z_o \text{ ohms} \qquad (2)$$

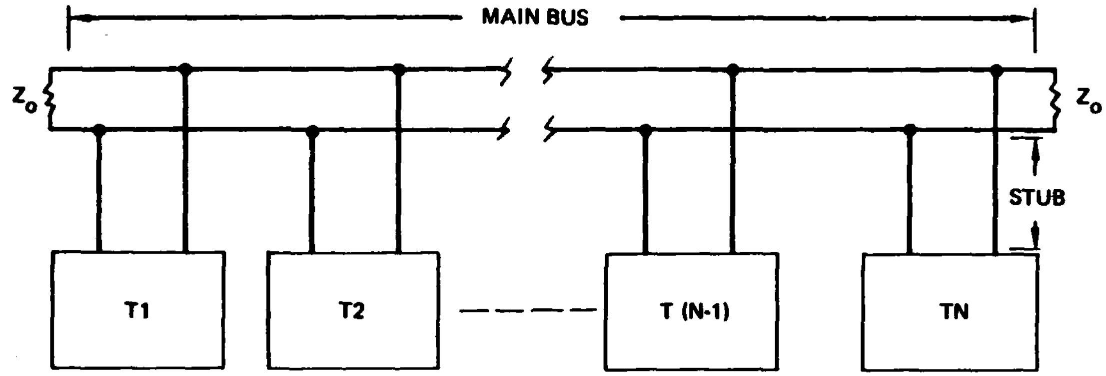


Figure 4.2-1. Bus Network Configuration

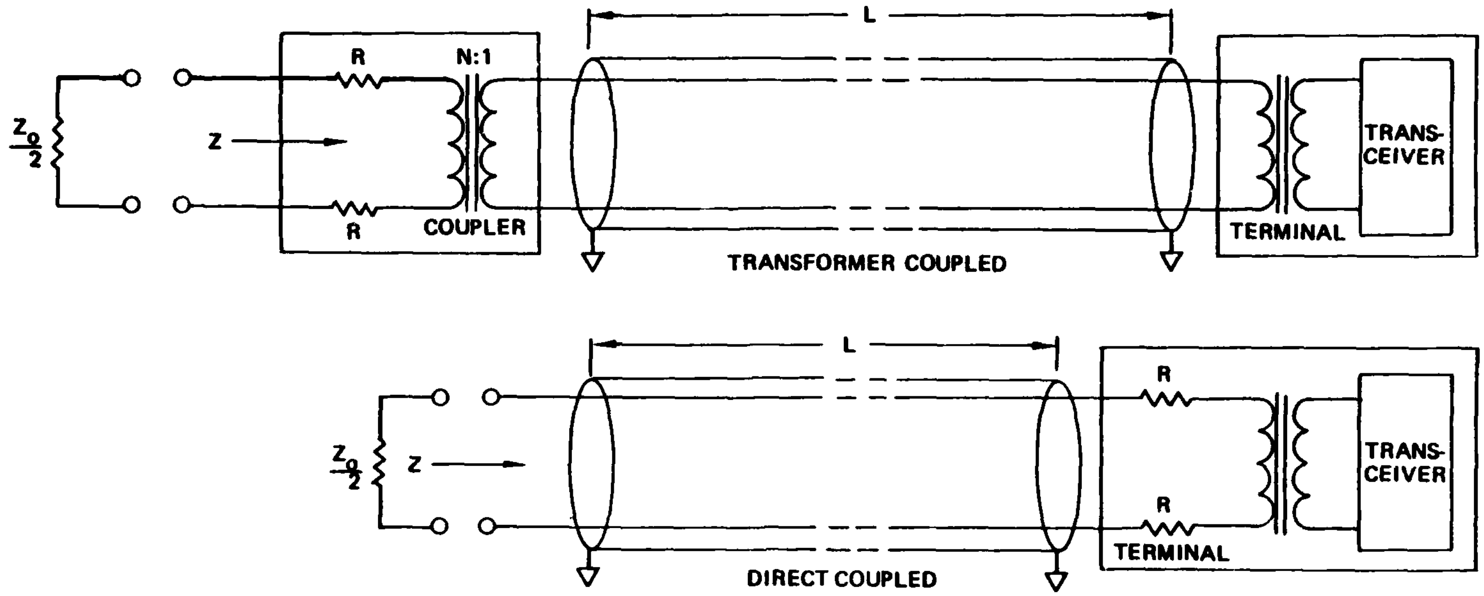


Figure 4.2-2. Transformer-Coupled and Direct-Coupled Stubs

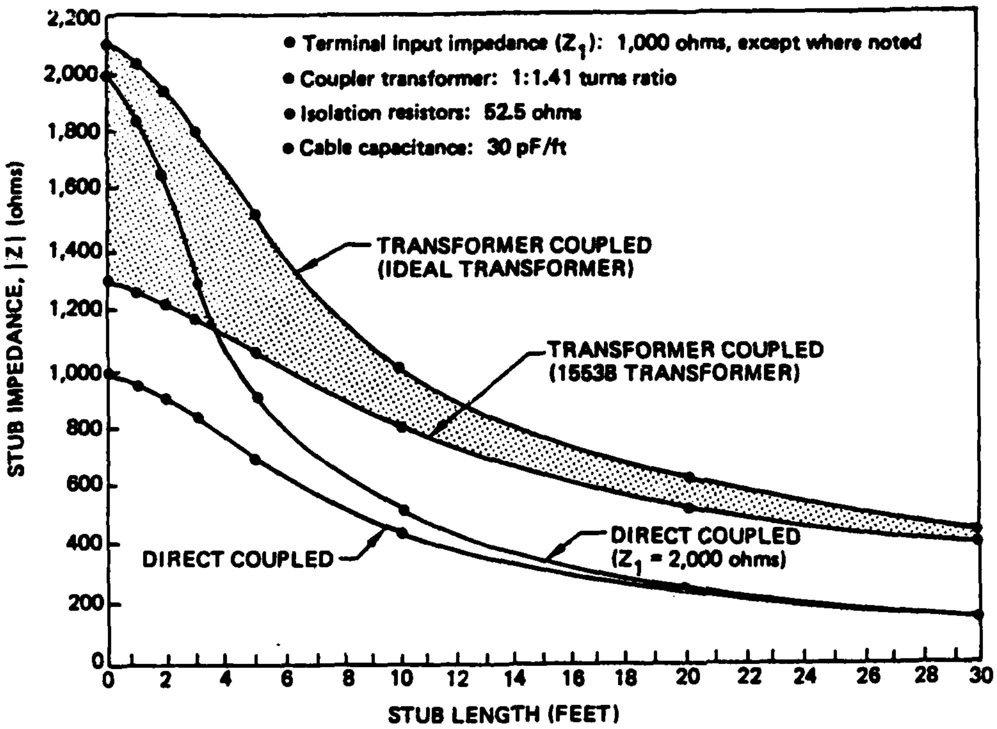


Figure 4.2-3. Stub Impedance Versus Length

The reflected impedance, Z_R, from the bus to the stub because of the transformer impedance transformation is:

$$Z_R = \frac{Z_B}{(1.41)^2} = \frac{2Z_o}{2} = Z_o \qquad (3)$$

Therefore, the coupler transformer specified in 1553B provides the characteristics desired for reducing reflections and maintaining signal levels for systems where long stubs are required. Direct coupling can be used for connections using stubs of 3 ft or less if terminal input impedance is maintained at the specified value.

Operational systems have been successfully implemented using 1:1 turns ratio coupler transformers. Many configurations can be built reliably if careful attention is paid to the number, length, and location of the stubs on the main bus. It is highly desirable to test a proposed network using a computer simulation or laboratory test setup. The computer-generated data bus simulation provides more flexibility during the early design stages. A number of bus network simulation programs have been developed with varying degrees of success.

4.2.1.2 Terminal Interface

An additional concern of the system integrator and the terminal hardware designer is the specification for the bus and terminal interface. This area of 1553A was significantly reworked to provide a more complete definition of the terminal interface characteristics that are independent of network configuration. Figures 4.2-4 and 4.2-5 show the interface diagrams and points of terminal signal measurement defined in 1553A and 1553B. Table 4.2-2 is a summary listing of the terminal and data bus interface requirements specified in the two versions of the standard (see fig. 4.2-6). The following discussion will relate some of the rationale for this approach to development of the updated specifications in 1553B.

Output Voltage

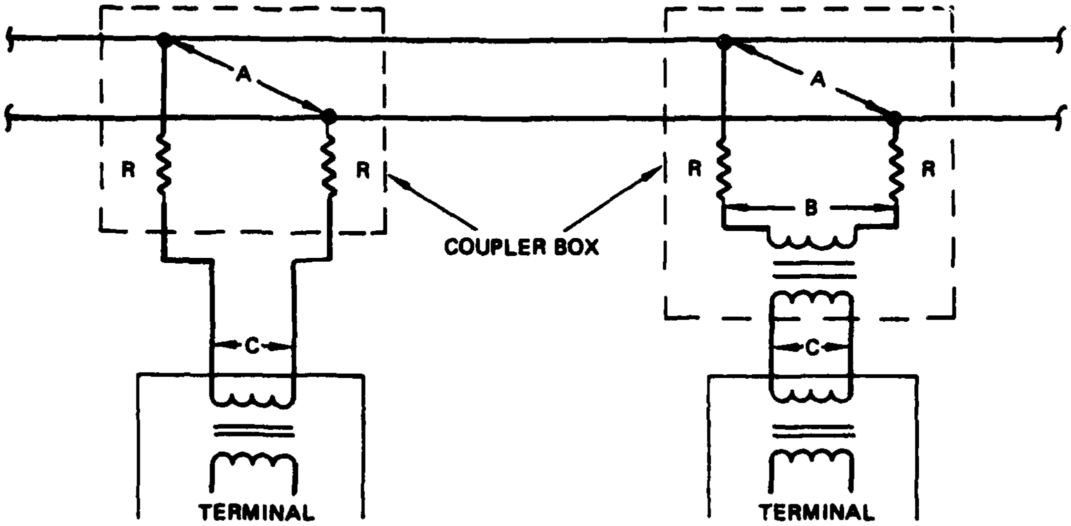
The upper end of the bus voltage range (20V p-p) allowed by 1553A is considered to be excessive and if implemented results in excessive power dissipation. Most of the systems and hardware designed to 1553A use signal levels at or near the lower end (6.0V p-p) of the specified range. It should be noted that the measurement point in 1553A is at the main bus, point A on figure 4.2-4. This does not provide a specified level at the terminal connection point (C) and is especially troublesome to the terminal hardware designer because the characteristics of the coupler transformer are not specified. The approach taken for 1553B is to specify the terminal output for the two conditions, transformer-coupled and direct-coupled. The coupler-transformer characteristics are specified. The turns ratio, 1:N, shown in fig. 4.2-5(b) is specified to be 1:1.41 for the reasons discussed in section 4.2.1.1. The 18V to 27V p-p transmitter output applied to the stub and coupler results in a nominal 6.0V to 9.0V p-p signal level at the stub and bus connection (point B). This range is equivalent to that specified for the direct coupled case shown in figure 4.2.5(a).

Output Waveform

The transmitted waveform specified in 1553A is limited in the definition of signals that appear on the data bus. Zero crossing deviations allowed are not well defined for all possible patterns, and the rise and fall times specification is open ended. The waveform characteristics defined in 1553B provide control of the zero crossing deviations for all possible conditions and establishes a limit on distortion.

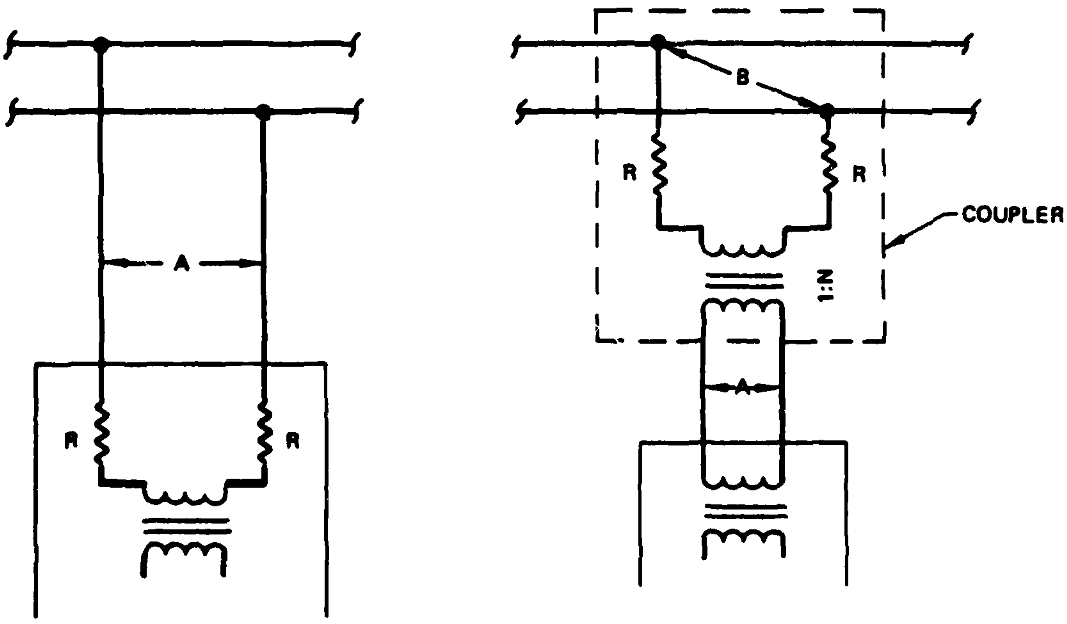
A significant debate has developed over the most desirable waveform characteristics for the transmitted data bus signal. The 1553A standard limits the rise and fall time to no less than 100 ns and 1553B specifies a range from 100 to 300 ns. Some aircraft programs have defined a limit on the harmonic content of the output to essentially restrict the waveform to a sine wave. This is in contrast to most programs, which permit a trapezoidal waveform with limited rise and fall times and limited droop. The transmission of a sine wave on the bus requires extensive filtering and implementation of a linear driver resulting in increased complexity and cost and a significant increase in output driver power dissipation.

It has also been found that a practical filter implementation that allows the rolloff characteristics results in an overshoot larger than that specified in paragraphs 4.5.2.1.1.2 and 4.5.2.2.1.2 of 1553B.



R Isolation resistor

Figure 4.2-4. MIL-STD-1553A Data Bus Interface



a. Direct Coupled (Short Stub) b. Transformer Coupled (Long Stub)

R Isolation resistor

N Transformer turns ratio

Figure 4.2-5. MIL-STD-1553B Data Bus Interface

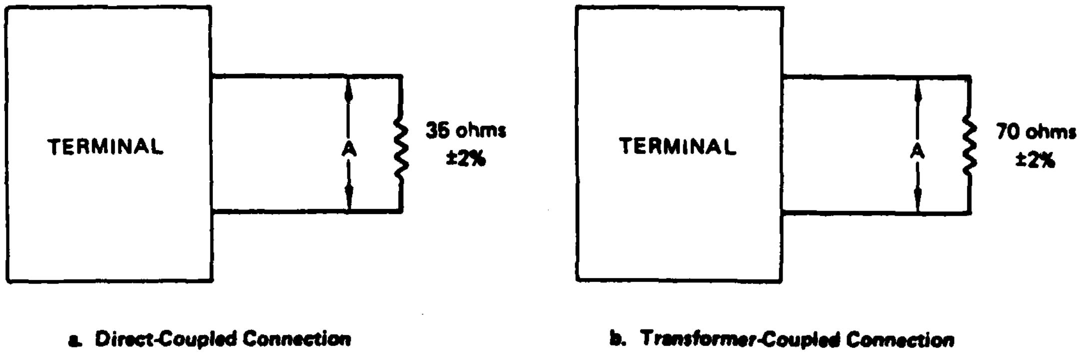


Figure 4.2-6. Direct-Coupled and Transformer-Coupled Terminal Output Test Configuration

The transmitter efficiency is reduced, resulting in increased dissipation for the same delivered power. The rationale developed to justify this approach is as follows:

a. The radiated harmonics of the unfiltered trapezoid can interfere with other equipment on the aircraft.

b. The mismatch inherent in the data bus network caused by nonideal terminations, stubbing and cable characteristics results in complex reflections because of the harmonic content of the unfiltered waveform.

EMI testing has been performed to measure the radiated interference from a twisted-shielded pair with 15V p-p into 50 ohms. The test was conducted in a shielded room, with the test cable penetrating the wall of the screen room and the shield grounded at the point of penetration. The 1553 waveform generator was located outside the screen room. Measurement techniques were in accordance with the procedures set forth in MIL-I-6181D. With a balanced drive and care taken to ensure that no significant common mode signal was impressed on the line, the interference level could not be detected above the receiver ambient noise level.

It is known by those familiar with 1553 data bus systems that the twisted pair shielded cable is essentially a distributed low-pass filter and the problem of item b is significantly reduced because a few feet of cable effectively provides a filtering effect.

The conclusion is that special filtering at the transmitter can be employed to reduce signal distortion and emanations from the bus if the added expense and complexity can be justified.

<u>Symmetry</u>

Symmetry of the transmitted waveform in time and amplitude is not adequately specified in 1553A. An ideal waveform is perfectly balanced so that the signal energy on both sides of zero (off) level is identical. If the positive or negative energy is not equal, problems can develop in the coupling transformers and the transmission line can acquire a charge that

Table 4.2-2. Summary of Terminal and Data Bus Interface Requirements

Parameter	MIL-STD-1553A	MIL-STD-1553B
• Terminal output characteristics		
• Output voltage	±3.0V to ±10.0V, peak, l-l (6.0V to 20.0V, p-p, l-l) Point A, figure 4.2-4	18.0V to 27.0V, p-p, l-l (transformer coupled) Point A, figure 4.2-6B 6.0V to 9.0V, p-p, l-l (direct coupled) Point A, figure 4.2-6A
• Output waveform	Zero crossing deviation = ±25 ns; rise and fall time of waveform shall be ≥100 ns	Zero crossing deviation = ±25-ns maximum, measured with respect to previous crossing; rise and fall times are 100 to 300 ns Overshoot and ringing = ±900 mV, peak, maximum, l-l Point A, figure 4.2-6B (transformer-coupled stub) Zero crossing deviation and rise and fall times same as above overshoot and ringing; ±300-mV peak, maximum, l-l Point A, figure 4.2-6A (direct-coupled stub)

appears as a tail with overshoot and ringing when transmission is terminated. These considerations require that the symmetry of the transmitted waveform be controlled within close practical limits. This is accomplished in 1553B by specifying the signal level from a time beginning 2.5 us after the midbit zero crossing of the parity bit of the last word in a message transmitted by the terminal under test. The test messages contain the maximum number of words and defined bit patterns. This consideration provides another argument for implementing the simplest transmitter possible, namely the trapezoidal driver, because the problem of maintaining a balanced drive condition is worsened for the highly filtered linear transmitter.

Output Noise

The originally specified value of 10 mV p-p noise is considered unrealistically low for practical hardware design. Noise is normally specified as an rms value because peak noise is difficult to measure. The output rms noise for the transformer-coupled and direct-coupled cases is specified in 1553B and is consistent with the required system performance and practical terminal hardware design.

Input Voltage

The input voltage specifications in 1553B have been revised to reflect the output voltage ranges for the transformer-coupled and direct-coupled

Table 4.2-2. Summary of Terminal and Data Bus Interface Requirements (Continued)

Parameter	MIL-STD-1553A	MIL-STD-1553B
• Output symmetry	Not specified	Voltage at 2.5 μs after midpoint of parity bit = ±250 mV, peak, maximum, l-l Point A, figure 4.2-6B (transformer-coupled stub) Voltage at 2.5 μs after midpoint of parity bit = ±90 mV, peak, maximum, l-l Point A, figure 4.2-6A (direct coupled stub)
• Output noise	10 mV, p-p, l-l Point A, figure 4.2-4	14.0 mV, rms, l-l Point A, figure 4.2-6B (transformer coupled) 5.0 mV, rms, l-l Point A, figure 4.2-6A (direct coupled)
• Terminal input characteristics		
• Input voltage	±0.5V to ±10.0V peak, l-l (1.0V to 20.0V, p-p, l-l) Point A, figure 4.2-4, terminal responds	0.86V to 14.0V, p-p, l-l, terminal response required; 0.0V to 0.2V, p-p, l-l, terminal no response (with transformer-coupled stubs) Point A, figure 4.2-5B 1.2V to 20.0V, p-p, l-l, terminal response required; 0.0V to 0.28V, p-p, l-l, terminal no response (with direct-coupled stubs) Point A, figure 4.2-5A

connections to the terminal. The terminal-required response and no-response signal levels are specified so that the optimum threshold levels may be selected. It should be noted that the threshold setting has a significant effect on the noise rejection and error rate performance of the receiver. The maximum "not to respond" signal level of 200 and 280 mV p-p allows optimum threshold settings of ±125 and ±175 mV, respectively, for minimum bit error rate (BER) performance when subjected to the specified noise rejection test conditions.

Input Impedance

As indicated in the data bus network discussion in section 4.2.1.1, input impedance is required to be maintained at a reasonable level to reduce the signal distortion effects when terminals are connected to the bus. Terminal input impedance is determined primarily by the following:

a. Transformer inductance and interwinding capacitance.

Table 4.2-2. Summary of Terminal and Data Bus Interface Requirements (Concluded)

Parameter	MIL-STD-1553A	MIL-STD-1553B
• Input impedance	2,000 ohms, minimum, from 100 kHz to 1.0 MHz Point C, figure 4.2-4	1,000 ohms, minimum, from 75 kHz to 1.0 MHz Point A, figure 4.2-5B (transformer-coupled stub)
		2,000 ohms, minimum, from 75 kHz to 1.0 MHz, Point A, figure 4.2-5A (direct coupled)
• Noise rejection	BER = 1 in 10^{12}, maximum Incomplete message rate 1 in 10^{6} Test condition—bus controller connected to RT over 100-ft data bus using 20-ft stubs	Maximum word error rate of 1 in 10^{7} with AWG noise = 1.0 kHz to 4.0 MHz, 140 mV, rms level Signal level = 2.1V, p-p, l-l Point A, figure 4.2-5B (transformer-coupled stub)
		Maximum word error rate of 1 in 10^{7} with AWG noise = 1.0 kHz to 4.0 MHz, 200 mV, rms level Signal level is 3.0V, p-p, l-l Point A, figure 4.2-5A (direct-coupled stub)
• Common mode rejection	±10.0V peak, line to ground, dc to 2 MHz, shall not degrade performance Point A, figure 4.2-4	±10.0V peak, line to ground, dc to 2.0 MHz, shall not degrade performance of the receiver Point A, figure 4.2-5B (transformer-coupled stub)
		Same specification for direct-coupled stub Point A, figure 4.2-5A

b. Stray capacitance from terminal wiring between the terminal connector and the receiver.

c. One-half of the impedance on the secondary is reflected to the terminal's input in the transformer-coupled case, because of the 1:1.41 turns ratio.

The factor of 2 difference in the impedance specified for the transformer-coupled and direct-coupled cases is based primarily on the effect of item c above. The frequency range was changed to reduce the lower frequency limit from 100 to 75 kHz. This provides additional assurance that adequate transformer volt-time product (inductance) is available to support the lower frequencies of the signal without approaching saturation.

Noise Rejection

The noise rejection specification and test conditions defined in 1553A require extensive system-type evaluation testing of the terminal employing a bus controller and data bus radiated with certain of the EMI fields specified in MIL-STD-461 and -462. Extensive test time is required to verify a BER of 10^{-12} and the test must be performed in a screen room.

The test conditions of signal and noise specified in 1553B were selected to produce a corresponding value of word error ratio (WER) that is sufficiently high (10^{-7}) to permit performance verification of a terminal receiver within a reasonable test period. The noise rejection is a figure-of-merit test and can be performed in a normal laboratory environment with a typical test setup as shown in figure 4.2-7. The verification of detector performance should consider the measurement of both detected and undetected errors. To measure undetected errors that do not correlate with the transmitted signal and are not detected by the terminal under test, it is necessary to compare the transmitted and received data. Therefore, a reference of transmitted data is provided to the comparator for comparison with the detected data from the terminal under test. Some considerations for designing the test setup and performing the test are related in a technical paper entitled "1553B: How Do You Know You'll Pass Noise Test?" presented by R. M. Salter at the AFSC Data Bus Conference in October 1978.

4.2.1.3 Data Bus Signal and Noise Considerations

Any unwanted interference that reduces the signal detection capability and increases the BER at the bus receivers is categorized as noise. Noise in a 1553 data bus system is the result of the following conditions:

a. Signal waveform distortion (internal)
b. Externally induced interference

The nonlinear characteristics of the twisted-shielded pair and reflections caused by nonideal termination and stubbing of the line result in distortion

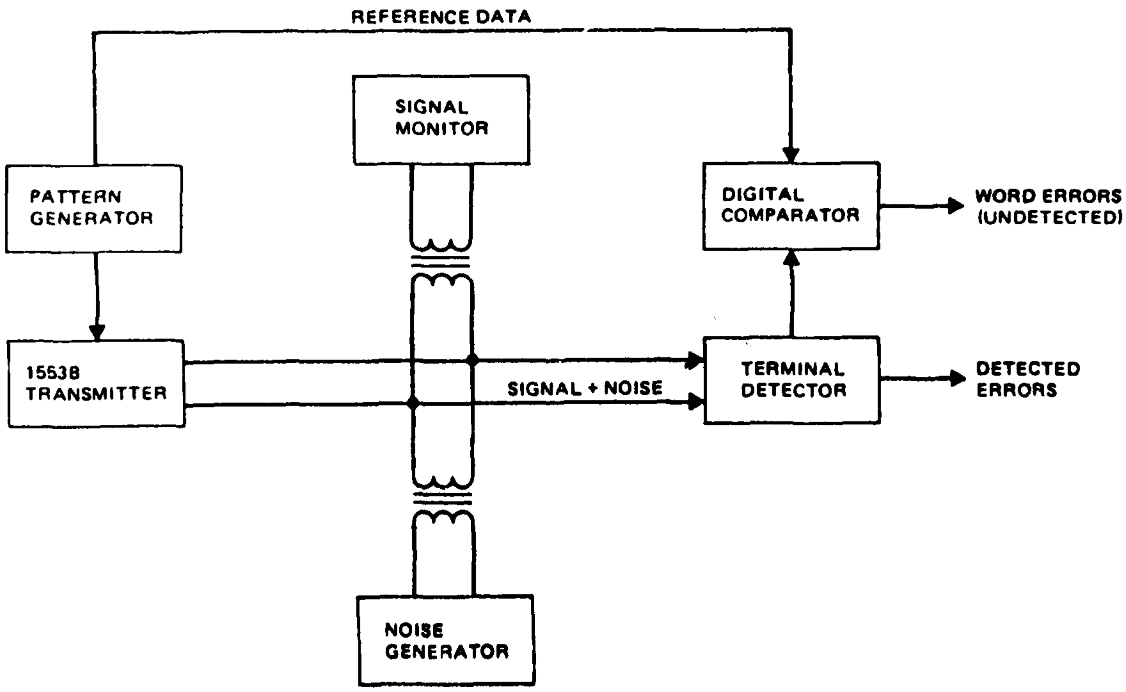


Figure 4.2-7. Typical Noise Rejection Test Setup

of the transmitted waveform. Phase distortion can result in intersymbol interference where bit transitions are not equally spaced in time. The major cause of intersymbol interference is the nonlinear characteristics of the data bus resulting in exponential decay of a bit, which can affect transition times and adjacent bit amplitude. As discussed in section 4.2.1.1, the control of bus nonlinearity and reflections includes minimizing the total line length and the use of stubbing and maximizing the stub impedance reflected to the bus.

Externally generated noise that may be induced on the data bus affects the detection error rates. The 1553 bus definition, employing transformer-coupled twisted-shielded pair transmission line and biphase signaling minimizes the effect of external noise. It is important to ensure that implementation of system wiring, especially the shielding and grounding at terminal points, does not negate the inherent immunity to external noise.

Externally generated noise on board an aircraft can take many forms with a wide variety of power and frequencies. It is recognized that impulse noise having either random or periodic impulse duration, frequency of occurrence, and burst interval are more typical of noise sources that have major impact on aircraft digital data systems. Relay switching is generally regarded as the most severe source of impulse noise on a typical aircraft. This type of noise defies accepted forms of analysis, such as that performed using additive white gaussian (AWG) noise model. Because of the difficulty of error performance analysis using the impulsive noise model, a worst-case gaussian model has been formulated. This model offers an analysis and test tool for evaluation of terminal receiver performance considering the effects of impulsive noise. This approach is reflected in the noise rejection test conditions and word error rate versus signal-to-noise ratio (SNR) performance requirements of 1553B, paragraphs 4.5.2.1.2.4 and 4.5.2.2.2.4. It should be emphasized that the bit error performance of the terminal receiver can be significantly improved by the application of a properly designed predetection filter that attenuates interfering signals without significant impact on intersymbol interference. A filter of this type is discussed in section 4.4.2.1.

The following is a list of the important considerations for minimizing the effects of externally induced noise:

a. Cable routing and length
b. Shield grounding and termination
c. Connector types and installation
d. Maintaining balanced line

It is obvious that the bus network cable routing and configuration should be chosen to provide maximum separation from potential interfering sources, such as power lines and control lines to inductive loads. The number and length of stubs should be minimized as a first step with the length of cable reduced to a minimum.

Shields should be carried through splices and connector breakpoints whenever possible. The shield should be grounded at every cable breakpoint using the shortest grounding strap possible. Connector types should be selected to allow a continuous shield through the connector wherever possible. Using a pin on a multipin connector for shield connection is not recommended. One

of the most important considerations for maximum immunity to induced noise is the balanced data bus. The twisted pair with transformer coupling and a balanced biphase signaling offers a distinct advantage in maintaining a high common mode rejection ratio (CMRR). For these advantages to be maintained in a practical installation, certain areas that impact the balanced condition should be considered. Some of these are --

a. Selection of cable type
b. Design of coupling transformers
c. Symmetry and proximity of signal pair
d. Transmitter drive

The cable physical and electrical characteristics are important to maintain the balance condition. The uniformity of the cable will affect the relative value of capacitance from each signal line to shield. The coupling transformer is a very important element of the signal channel. The characteristic of shunt impedance at both high and low frequencies has significant impact on the cable loading and signal waveform integrity. The interwinding capacitance determines the common mode rejection capability of the transformer. The minimum values of the coupler-transformer characteristics are specified in 1553B. The coupling transformers used in the separate line coupler and internal to the terminal are of equal concern. Design considerations and guidelines are included in sections 4.3.1 and 4.4.1.

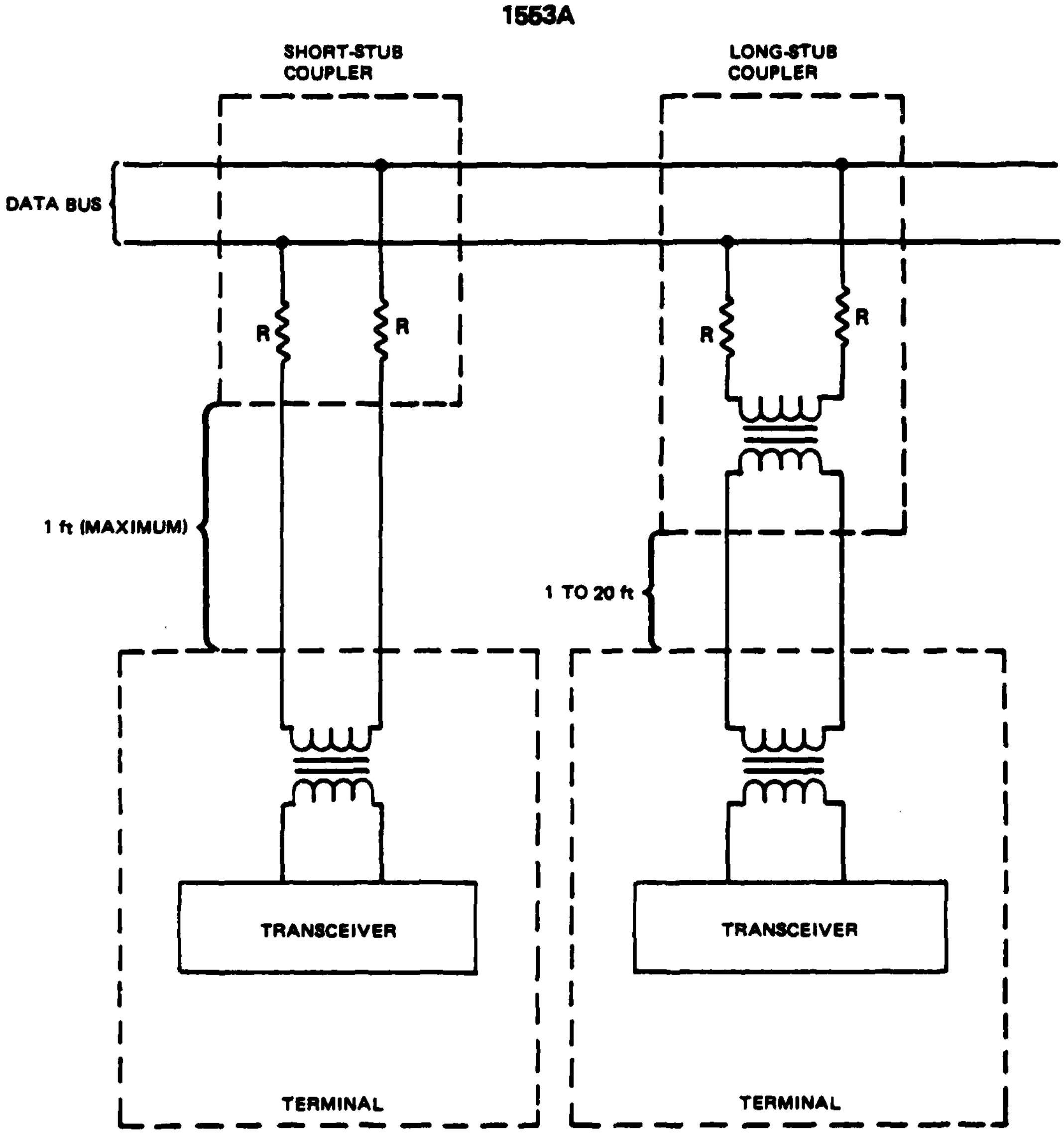
There has been debate over the issue of grounded center taps on the line side of the coupling transformers. Most system designs do not include the center tap. One major aircraft program employs this technique for induced noise cancellation. The related research shows an improved CMRR because of the balancing effects of the center-tapped winding.

4.3 BUS COUPLER DESIGN

Bus coupler networks, separate from the terminals, are required by 1553B when connection to the data bus via "long stubs" is required. A long stub is defined to be greater than 1 ft. Direct coupling, which can be implemented without a separate coupler box, is defined for short-stub connections of 1 ft or less. The long-stub coupler network incorporates isolation resistors and a coupling transformer.

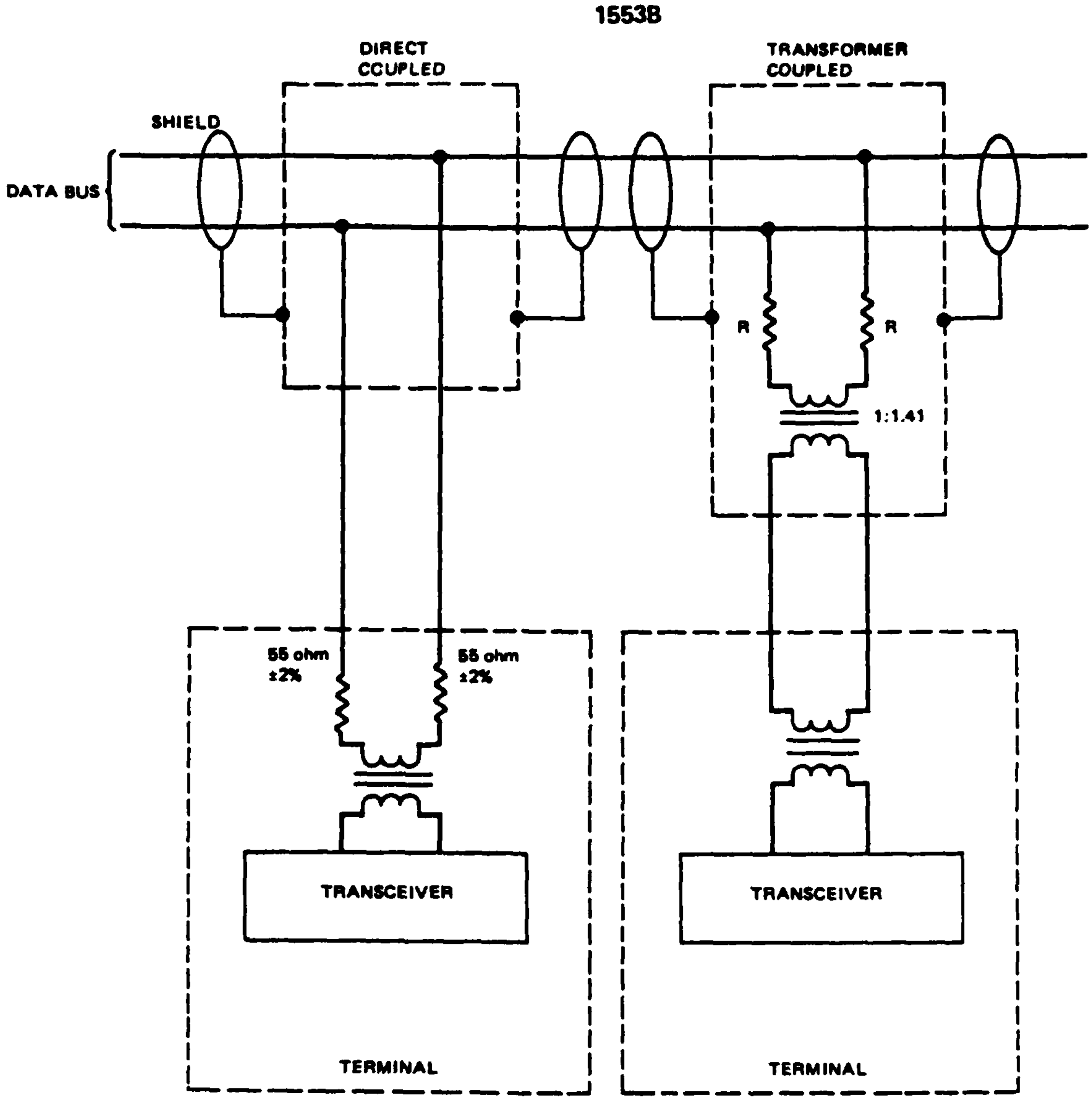
The requirements for couplers specified in 1553A have been modified for 1553B and a comparison of the two requirements is shown in figure 4.3-1. The major differences in the two requirements are the placement of the isolation resistors for the direct-coupled (short-stub) connection and the characterization of the coupling transformer in the long-stub (transformer-coupled) connection. With the isolation resistors located in the terminal for the direct-coupled case, the need for a separate coupler box is eliminated as long as a reliable shielded splice can be made. In most cases, the bus connections can be spliced in the cable connector that mates with the terminal connector.

The coupler-transformer characteristics are very important to the signal integrity and noise performance of the data bus system. The purposes of the coupler are to (1) provide isolation of the main bus for fault conditions on the stub or in the terminal, (2) provide some reduced bus signal distortion



- Isolation resistors: $R = 0.75\ Z_o^* \pm 5\%$
- Isolation transformer: (not specified)
- *Nominal characteristic impedance of bus cable: $Z_o = 70 \pm 10\%$ at 1 MHz

Figure 4.3-1. Coupler Characteristics



- Isolation resistors: $R = 0.75\ Z_o^* \pm 2\%$
- Isolation transformer: turns ratio 1:1.41 ±3%
  (1—terminal winding)
  (1.41—bus winding)

  $Z_{oc} > 3K$ at 75 kHz to 1 MHz
  1V rms sine wave

  Droop: < 20%
  Overshooting/ringing: < ±1V
  CMR: > 45 dB at 1 MHz
  } at 27v P-P 250-kHz square wave

- Nominal characteristic impedance of bus cable:
  Z_o = 70 to 85 at 1 MHz

Figure 4.3-1. Coupler Characteristics (Continued)

effects by increasing the effective stub impedance, and (3) provide termination of the stub when transmitting from the terminal. The isolation resistors and the transformer turns ratio 1:1.41 provide the benefits listed above. The terminal input and output speci ations for the transformer-coupled (coupler) and direct-coupled connect. .s are required to be separated in 1553B because of the effects on signal levels and impedances of the transformer turns ratio being specified as 1:1.41 instead of 1:1. Refer to section 4.2.1.1 for a detailed treatment of the coupler and stub considerations related to bus network design.

4.3.1 Transformer Characteristics

The section presents some of the major coupler-transformer design considerations. The generalized design approach also applies to the coupling transformer used in the terminal transceiver, which is described in section 4.4.

4.3.1.1 General Design Considerations

The use of transformers as a means of coupling baseband signals in a balanced transmission line system has proved to be an extremely effective approach. When designed and used properly, transformers can perform effectively to maintain isolation and signal integrity at a relatively low cost and with high reliability. They are not extremely lossy and can readily achieve high common mode rejection and impedance transformation for circuit compatibility. A poorly designed coupling transformer can cause significant problems in signal distortion, line reflections, and high bit error rates. It is appropriate to discuss some of the pertinent coupler-transformer design considerations and trade-offs.

It is convenient to represent pulse or broadband transformers in terms of two equivalent circuits; the high-frequency equivalent circuit is shown in figure 4.3-2(a) and the low frequency circuit is shown in figure 4.3-2(b). The upper frequency cutoff, f_2, of a critically damped transformer is given by the equation:

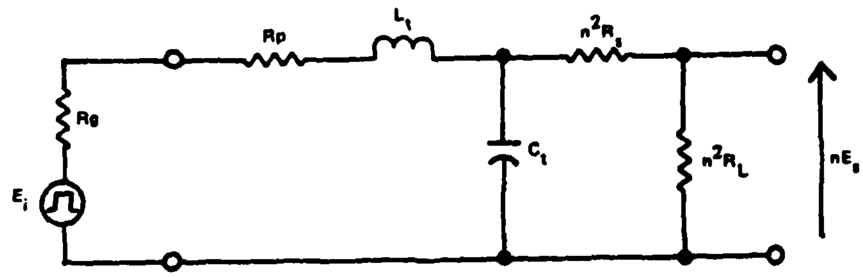
$$f_2 = \frac{1}{2\pi\sqrt{\alpha\, L_t C_t}} \tag{1}$$

where,

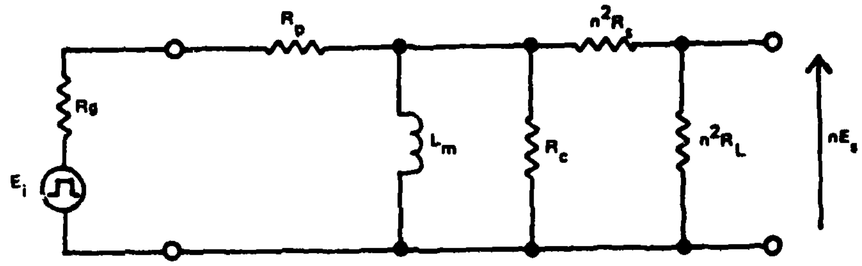
L_t is the leakage inductance
C_t is the equivalent short circuit capacitance
α is the attenuation constant

The values of L_t and C_t are determined by the geometry of the windings, the mean diameter of the core, the dielectric constant of the insulation and the turns ratio. The relationship between rise time, t_r, and the upper frequency response is

$$f_2 = \frac{0.35}{t_r} \tag{2}$$



a. High-Frequency Equivalent



b. Low-Frequency Equivalent

Figure 4.3-2. Transformer Equivalent Circuit

The *low-frequency cutoff* is approximately

$$f_1 = \frac{R}{2\,\pi\,L_m} \tag{3}$$

where,

L_m is the primary inductance, and R is a combined resistance of the resistors shown in the equivalent circuit.

where,

XL>>R, the equation becomes

$$T_1 = \frac{XL}{2\,\pi\,L_m} \tag{4}$$

A useful expression for primary inductance, L_m, which will aid in determining the parameters of the transformer based on the selected core is:

$$L_m = 0.4\,\pi\,\frac{AcuNp^2}{\ell}\ X\ 10^{-8}\ H \tag{5}$$

where,

Ac = core cross sectional area in centimetres
u = permeability of the core material
Np = number of turns on the primary
ℓ = mean magnetic path length in centimetres

and an approximation for the rise time, t_r, is

$$t_r = \frac{6NpD}{n} \sqrt{\alpha K} \times 10^{-10} \text{ sec}$$

where,

D = the mean diameter of the core
K = the dielectric constant
n = turns ratio

Using the equivalent circuits and simplified equations listed above, a set of coupling-transformer parameter relationships can be developed. Table 4.3-1 is a summary of the more significant parameters affecting the transformer lower and upper frequency response. The following is a summary description of the pertinent relationships and design trade-offs required in a practical transformer design:

a. As shown in the table, the predominant parameter affecting the lower frequency response, f_1, is magnetizing inductance, L_m. Increasing L_m results in decreased high-frequency response.

b. Magnetizing inductance, L_m, is a function of the square of the turns, N_p, and is directly proportional to permeability, u, and cross-sectional area, A_c, of the core and inversely proportional to the mean magnetic path length, ℓ, or mean diameter, D. This indicates that a minimum number of turns, low u core material, and a small core are desired for high-frequency response.

c. Increasing the length of wire as the number of turns is increased or decreasing the size of the wire results in higher resistance, R, which tends to improve the lower frequency response at the expense of increased signal losses.

d. In most cases, where parameters affect both the high- and low-frequency response, the results are in the same direction. In other words, if a parameter lowers the lower frequency response, f_1, that same parameter also lowers the upper frequency response. An example of this is the number of turns parameter, N, shown in table 4.3-1 as a contributing factor for both f_1 and f_2. For f_1, L_m is proportional to the square of the turns, N, while C_t and L_t are proportional to the number of turns. The effect in both cases is a reduced frequency response that tends to impact the lower frequency cutoff more than the upper frequency.

e. The predominant factors affecting the upper frequency response are leakage inductance, L_t, and shunt capacitance, C_t. Increasing these parameters lowers the frequency cutoff.

f. The leakage inductance, L_t, is proportional to the number of turns, permeability of the core material, cross-sectional area of the core and mean magnetic path length, ℓ, or mean core diameter, D. Increasing L_t by increasing any or all of these parameters results in lower upper frequency cutoff.

Table 4.3-1. Coupling Transformer Parameters and Relationship

Frequency response	Predominant parameter	Contributing parameters and relationship				
		Number of turns (N)	Core material (μ)	Mean magnetic path length	Resistance (R)	Dielectric constant (K)
Lower frequency response $f_1 \approx \frac{R}{2\pi L_m}$	Magnetizing inductance $L_m = 0.4\pi \frac{Ac\,\mu\,Np^2}{\ell}$	$N^2 L_m$	$\left.\begin{matrix}Ac\\ \mu\end{matrix}\right\} L_m$	$\left.\begin{matrix}\frac{1}{\ell}\\ \text{or}\\ \frac{1}{D}\end{matrix}\right\} L_m$	$R \rightarrow Rp \rightarrow N$	–
Upper frequency response (f_2) $f_2 = \frac{1}{2\pi\sqrt{\alpha L_t C_t}}$	Leakage inductance (L_t) Shunt capacitance (C_t)	$N \rightarrow C_t, L_t$	$\left.\begin{matrix}\mu\\ Ac\end{matrix}\right\} L_t$	$\left.\begin{matrix}\ell\\ \text{or}\\ D\end{matrix}\right\} L_t, C_t$	$\frac{1}{\alpha} \rightarrow Fp \rightarrow N$	$K \rightarrow C_t$

g. The shunt capacitance, C_t, is directly proportional to the number of turns, mean magnetic path length or mean core diameter, and the insulation dielectric constant, K. By increasing any of these parameters, a lower upper frequency cutoff results.

h. It is also important to construct the transformer using winding techniques that minimize the shunt and interwinding capacitance. The interwinding capacitance has a significant effect on the high-frequency common mode rejection (CMR).

The following is a discussion of the specific coupler-transformer requirements defined by 1553B. Guidelines for design, construction, and evaluation are presented to aid the designer. It is apparent that an optimum transformer for the data bus application requires extensive trade-offs in parameter selection and careful attention to construction details.

4.3.1.2 Turns Ratio

The transformer in the 1553B coupler has the turns ratio of 1:1.41. This ratio, together with the 0.75Z fault isolation resistor provides the correct characteristic impedance for terminating the stub:

$$Z\ stub = \left(\frac{1}{1.41}\right)^2 (0.75\ Z_o + 0.75\ Z_o + 0.5\ Z_o)$$

The stub capacitance is also effectively decreased by the square of the turns ratio to lessen the loading problem. The 1:1.41 ratio of 1553B is a compromise between stub matching and decreased stub loading. A higher turns ratio would improve the loading problem. However, the stub would no longer be terminated in the characterisitc impedance. A more detailed discussion of the stub matching problem is presented in section 4.2.1.1.

4.3.1.3 Open Circuit Impedance

The transformer open circuit impedance (Zoc) is required to be greater than 3 kΩ in 1553B systems. The measurement is made looking into the higher turns winding (1.41) with a 75 kHz to 1 MHz sine wave signal. The test amplitude at the transformer winding is adjusted to 1V rms. The critical factors in achieving the 3 kΩ Zoc is the distributed capacitance of the windings and the transformer primary inductance. The inductance of the transformer must be large enough to provide the open circuit impedance at 75 kHz while the distributed capacitance should be small enough to maintain the open circuit impedance at the 1 MHZ test frequency. Using the formulas in section 4.3.1.1, the minimum inductance is 6.37 mH and maximum distributed capacitance is 53 pF. The inductance may obviously be increased by increasing the number of turns on the transformer. This technique, however, tends to increase the distributed capacitance, degrading high frequency performance and therefore causing waveform integrity and common mode rejection to suffer. Techniques for minimizing the interwinding and core-to-winding capacitance are described in the following sections.

4.3.1.4 Waveform Integrity

The ability of the coupler transformer to provide a satisfactory signal is specified in the droop, overshoot, and ringing requirements of 1553B as shown in figure 4.3-3. Droop is specified at 20% maximum when driving the transformer with a 250 kHz, 27V p-p square wave. The test for the droop characteristic is made by driving the low turns winding through a 360 ohm resistor and measuring the signal at the open-circuited high side winding. The droop of the transformer is determined mainly by the primary inductance. Since the primary inductance also provides the 3 k open circuit impedance, the inductance should be made as high as possible without degrading the high-frequency performance of the transformer. High-frequency performance may be improved by lowering the total number of turns on the transformer. This requires the use of a high-permeability core material that allows the inductance to be kept high with fewer turns. A calculation of the cutoff, f, indicates a value of 50 kHz is required for the specified droop characteristic.

Ringing and overshoot on the transformer signal is shown in figure 4.3-3. The ±1V limit on these high-frequency perturbations can be achieved through careful attention to leakage inductance and transformer capacitance.

4.3.1.5 Common Mode Rejection

The CMR of the isolation transformer is required to be greater than 45 dB. The common mode test shown in figure 4.3-4 consists of driving the low turns winding while measuring the differential signal across the high side. CMR can be improved by minimizing the interwinding capacitance and the core-to-winding capacitance.

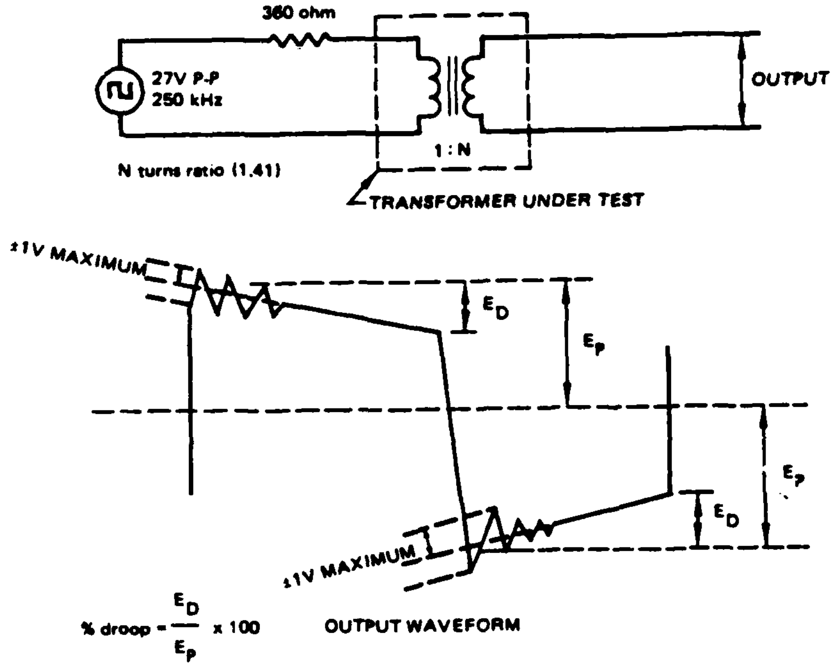


Figure 4.3-3. Waveform Test

Interwinding and core-to-winding capacitance may be reduced by reducing the total number of turns on the core. This requires the use of a high-permeability core material as described above. Core-to-winding capacitance can be further reduced by winding the core with an insulating tape such as nylon before winding. Interwinding capacitance can also be reduced by increasing the insulation (dielectric) thickness between windings. This technique may be limited in some applications because of physical size limitations.

A low number of turns with a high-permeability core will also decrease the leakage inductance of the transformer but will increase the possibility of core saturation at high current levels.

Another technique that may be useful in some applications is the segmented or "window winding" technique, wherein the primary and secondary windings do not overlap but are wound on opposite sides of the core. This technique reduces the interwinding capacitance and interwinding inductance but of course increases the size of the transformer and the labor involved in winding. For these reasons, segmented winding may not be desirable in applications where physical size or cost are limiting factors.

4.3.2 Packaging

The bus coupler, as a physical unit separate from the terminal hardware, presents some special problems to the system integrator and the aircraft electrical network designers. The coupler is ideally located near the main

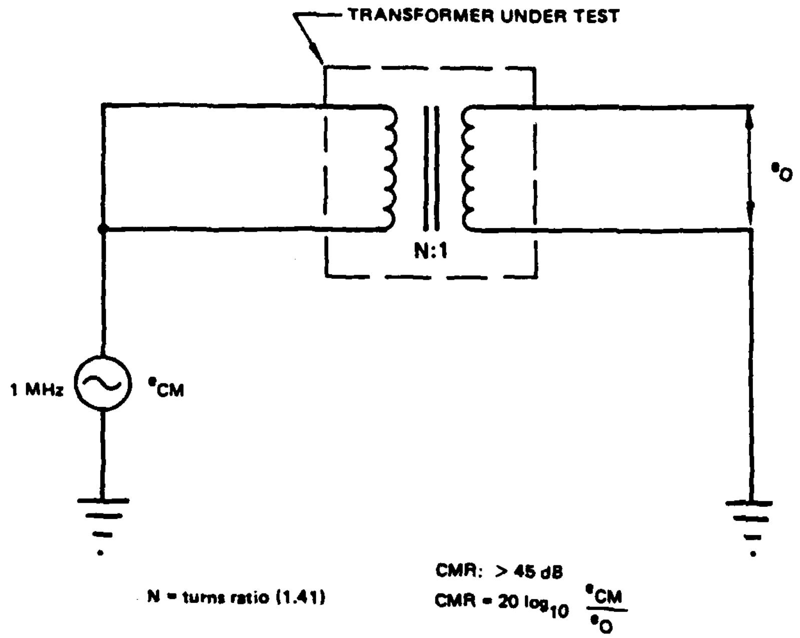


Figure 4.3-4. Common Mode Test

bus at the stub junction for the best electrical performance and fault isolation properties. This requires that a number of small line replaceable units (LRU) be mounted at various locations in the aircraft. The following is a list of some of the major factors that determine the coupler packaging:

a. Location in aircraft (environment and size)
b. Mounting
c. Number and type of connectors
d. Shielding and grounding
e. Number of coupler networks per package

A variety of coupler packages have been developed for existing programs. The F-16 aircraft avionics bus uses a small (1 in3) coupler box with twisted pair "pigtails" installed with other units in junction boxes. This unit, called a multiplex transfer network, is shown in the photograph of figure 4.3-5 on the lower left side. A similar-sized unit is used for stub coupling in the space shuttle data bus. Multiple couplers within a single housing have been proposed and used in some applications. Two- and three-connector standard data link couplers have been developed and produced for a variety of applications. Figure 4.3-5 shows some typical production couplers. Another unit, called a data link terminator designed for the AAH system, is a 1 in cube with a single multiple-pin connector.

The connector type specified is important for severe-environment military aircraft applications. MIL-STD-1553A specified the use of two-pin polarized connectors such as TEI BJ37 (reference to "TEL-14949-E137" is in error). The two-pin polarized connector employs an interface configuration with one male and one female contact. The female contact is imbedded in one side of

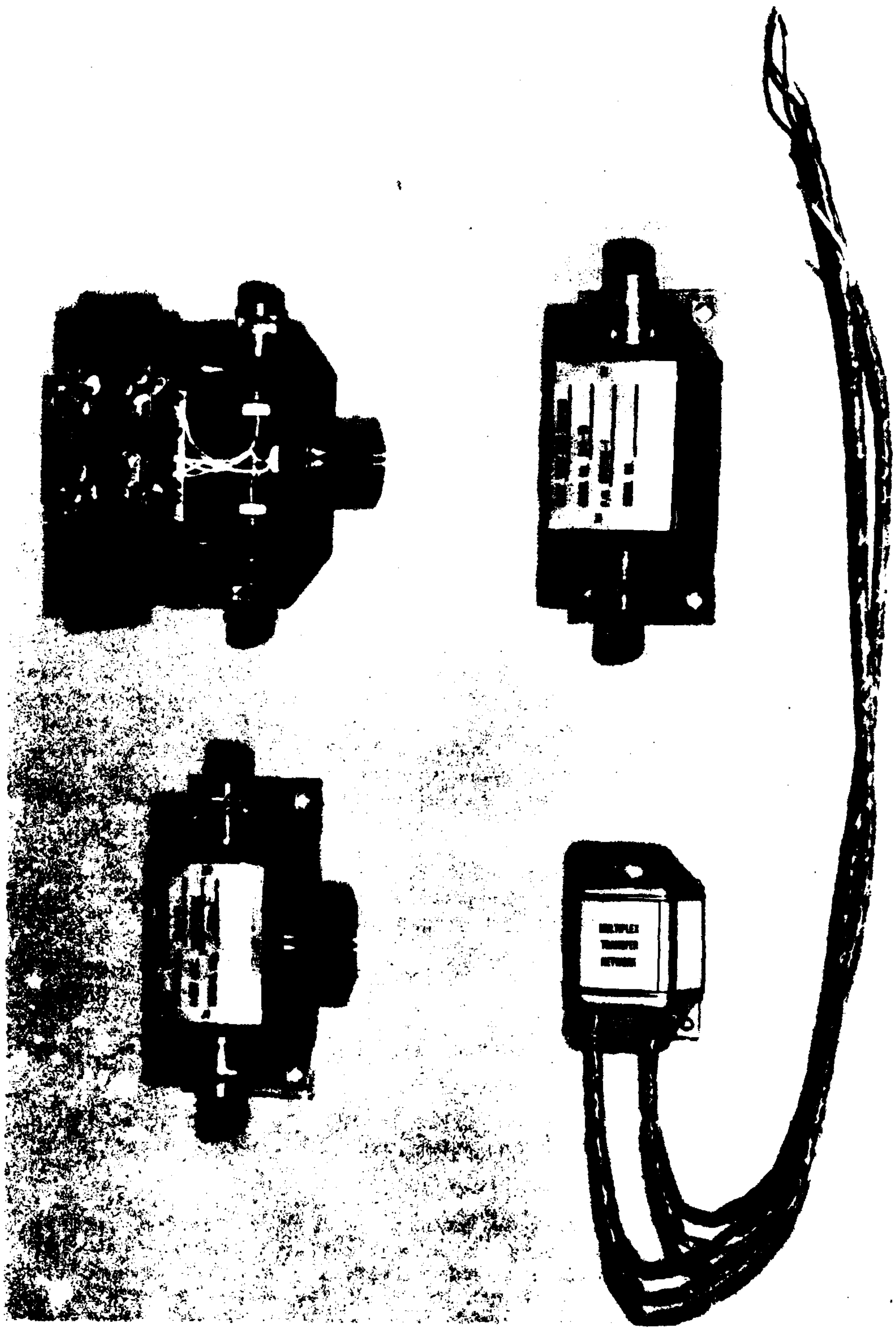


Figure 4.3-5. Typical MIL-STD-1553 Data Link Couplers

a step dielectric, and the male contact is exposed. The inherent shortcomings of this design include the following:

a. As stated in 1553A, "The polarity convention shall be that the female connection in the plug is positive, and the male connection in the receptacle is positive." This switching of polarity between plugs and receptacles is at best confusing and frequently results in reversal of polarity at installation.

b. Mating of two connectors requires physical alignment of the step dielectric. This results in continual wearing of the dielectric edge with each mating, and a high incidence of dielectric chipping and cracking has been observed.

c. The design of the step dielectric leaves one-half of the spring fingers (braid contacts) unsupported on the inner surface. Most manufacturers fabricate these spring fingers from half-hard brass, resulting in a design vulnerable to bending, breaking, or even shorting the male pin to ground. This deficiency can be minimized by using beryllium-copper spring fingers fully enclosed in a half-hard brass cylinder.

d. The available configurations of two-pin polarized connectors are limited in scope, and variations such as isolated ground bulkhead jacks, tees, cable entry jacks, and subminiature sizes are not readily available.

MIL-STD-1553B does not specify connector types. If BNC-type connectors are employed, the connector manufacturers encourage the use of concentric conductor connectors rather than two-pin polarized connectors. The internal configuration of these connectors consists of a center contact surrounded by an intermediate contact, another dielectric, and finally the shield contact. The front face of these connectors is identical for either twinax or triax applications and internally differ only slightly to accommodate either twinax or triax cables. The obvious advantages of the concentric design are --

a. The polarity convention is the same for a plug or a receptacle, minimizing the possibility of reversing polarity at installation. Normal polarity convention is for the center pin to be positive.

b. Mechanical alignment (radially) of the two connectors is not required for mating.

c. Spring fingers are fully supported by dielectric and far less susceptible to damage.

d. Concentric design connectors are available in BNC, C, or TPS sizes, in a variety of configurations. The connector type is usually dictated by the cable size.

Some recent applications for advanced military avionics systems specify subminiature pin type MIL-SPEC connectors meeting the requirements of MIL-C-38999. The increased ruggedness and reliability of these connectors is achieved at the expense of size and cost of the data bus couplers. Special care must be taken to ensure continuous shielding for EMI suppression. Cable shield may be carried through the connector using a

third pin even though this is not preferred. The metal case of the coupler should be sealed against moisture and EMI and connected to cable shields. Intimate contact of the coupler case with the aircraft frame ground is also required.

4.4 TRANSCEIVER DESIGN

The transceiver provides terminals with an interface to the data bus and performs the major functions of (1) bus signal coupling, (2) input signal coupling (from bus), (3) threshold detection, and (4) transmission of data from the encoder to the bus. Bus controllers and remote terminals employ the transceiver for connection to the data bus, while the monitor terminal may require only the receiver portion because it is not required to transmit. Figure 4.4-1 is a simplified diagram of a typical transceiver circuit showing the major functional elements.

4.4.1 Coupling Network

The coupling network provides bus connections for the transformer-coupled (external coupler) and direct-coupled cases defined in 1553B. Isolation resistors of 55 ohm value are included for the direct-coupled connection, and the proper transformer turns ratio is provided when the appropriate bus connection is selected. The turns ratio is different for the transformer-coupled and direct-coupled connections to compensate for the 1:1.41 reduction of signal level in the external coupler. This feature allows a threshold setting that is the same for both bus connections. The receiver transformer can be used for coupling the transmitter drivers to the bus with the addition of the center-tapped winding, as shown in figure 4.4-1.

The transformer is a very important element in determining the transceiver characteristics such as input impedance, signal waveform integrity, and common mode rejection required by 1553B. Considerations for transformer and associated input-output circuit design are as follows:

a. Provide the specified input impedance at high frequencies (terminal input impedance 1,000 and 2,000 ohms at 1 MHz).

b. Maintain waveform integrity and low percentage droop for the lower frequency conditions (less than 20% for 250 kHz square wave).

c. Design for low interwinding capacitance to achieve the common mode rejection (45 dB CMR at $\pm$10V peak, dc to 2.0 MHz).

These considerations, along with some tips for design and construction of the bus coupler transformer presented in section 4.3.1, are directly applicable to the design of the transceiver transformer. The construction of the transceiver coupling transformer is somewhat more complex because of the additional winding for the transmitter and the multiple bus connections.

In addition to the transformer characteristics, other considerations for maintaining the terminal minimum input impedance specified in 1553B are as follows:

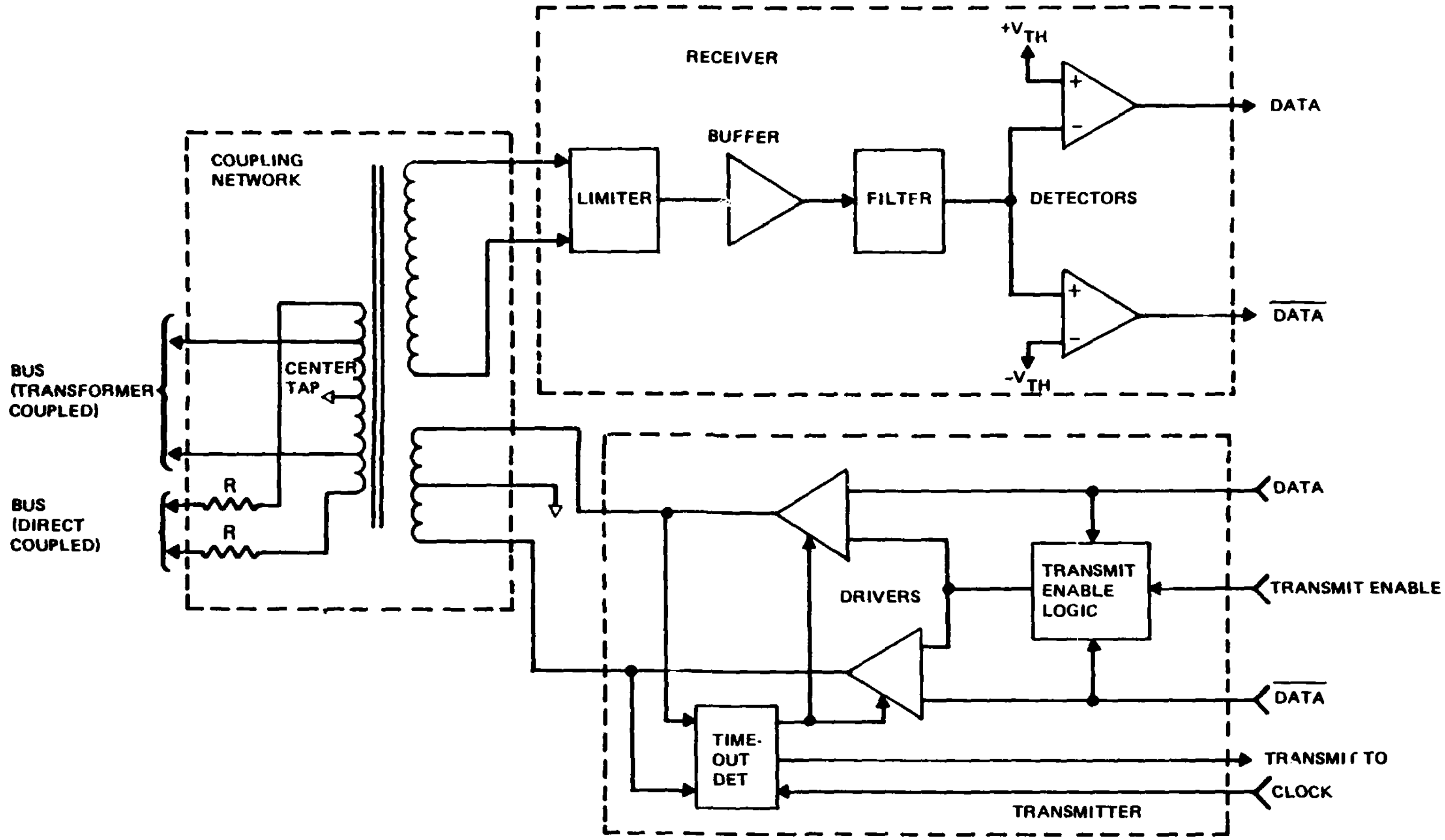


Figure 4.4-1. Typical Transceiver Circuit

a. Minimize stray capacitance of wiring from the external connector and on the circuit card to the buffer amplifier (every 100 pF results in 1600 ohms of shunt impedance).

b. Maintain high impedance at the receiver limiter and filter circuit inputs and transmitter driver outputs in the "off" state. These impedances must be maintained with the terminal (transceiver) power off.

The requirement for low output noise of 14 mV rms and 5 mV rms when not transmitting also places significant constraints on the length and routing of input-output wiring because of the induced power supply and logic noise generated in the terminal. There is a definite advantage to be gained in locating the coupler network and the receiver limiter and buffer amplifier as close as possible to the terminal input connector.

4.4.2 Receiver

The major functions of the receiver shown in figure 4.4-1 are (1) signal limiter, (2) buffer amplifier, (3) input filter, and (4) threshold detector. The limiter is required to clamp large signals at levels that can be accommodated at the buffer amplifier input. Large overvoltages can cause the amplifier to saturate, resulting in potential frequency response problems because of the increased amplifier recovery time. The buffer amplifier is employed to provide a high impedance reflected back to the bus and a low output impedance for driving the input filter. This buffer amplifier may not be required for all designs if proper impedance matching can be attained.

4.4.2.1 Input Filter

Selection of the input filter is an important design decision. The essential purpose of the predetection filter is to reduce the BER at the receiver output by improving the input SNR. Two types of noise have been considered in the investigation of 1553 type data bus systems: impulsive noise and AWG noise. Impulsive noise is the more difficult to deal with analytically and is the significant offender in this type of digital communication system. Impulsive noise measurements have been made on twisted-shielded pair cable, with an interfering line connected to a relay with worst-case noise characteristics. Evaluation of the spectrum shows a large concentration of noise power above 1.5 MHz and very little noise power below this frequency. Significant noise power can extend all the way up to 40 MHz in some cases. From these observations, it is apparent that a filter is required to reject high-frequency noise while passing the desired signal without introducing excessive intersymbol interference or signal attenuation. Some designers implement a bandpass active filter. It is [illegible] that a low-pass filter with linear phase in combination with the coupling transformer, which has a limited low-frequency response, provides a reasonable combination for most applications.

A conventional approach to the design of a filter that restricts the bandwidth of pulse data as much as possible without introducing excessive intersymbol interference is based on the so-called raised-cosine frequency characteristic. This type of filter is discussed by Bennett and Davey in "Data Transmission," chapters 5 and 7, and Lucky, Salz, and Weldon in "Principles of Data Communications," chapter 4. The scheme actually

consists of two filters, one at the transmitter and the other at the receiver. The product of the two filter transfer functions yields a raised-cosine amplitude spectrum. As discussed previously in section 4.2.1.2, filtering between the transmitter and the data bus for the purpose of reducing EMI or for improved performance in the presence of noise does not appear to be necessary or desirable. Therefore, the recommended approach is to insert a filter at the receiver input, which provides the response as described below.

The receiver filter transfer function was derived and an excellent approximation was determined, which has been tested and yields negligible intersymbol interference, producing a raised-cosine step response approximately as shown in figure 4.4-2. Figure 4.4-3 illustrates how well this filter provides the desired raised-cosine frequency response characteristic when driven by nonreturn to zero (NRZ) data. Figure 4.4-4 is a schematic of the synthesized three-pole raised-cosine filter circuit, which has been further simplified for practical implementation as shown in figure 4.4-5. Extensive evaluation has been conducted of the BER versus SNR performance of various biphase receivers employing this filter. The results show excellent performance based on analysis using worst case AWG noise.

Other filter types, such as active three-pole Butterworth, are also employed in many designs.

4.4.2.2 Threshold and Line Active Detectors

The filtered biphase is input to the threshold detectors, which are positive, and negative biased voltage comparators or slicers that provide an output when the input signal exceeds the preset threshold levels. Positive feedback is employed to provide hysteresis and ensure hard comparator decisions. The comparator outputs are provided to the biphase detector for further processing.

4.4.3 Transmitter

The transmitter shown in the transceiver circuit, figure 4.4-1, consists of current or voltage mode drivers operating in push-pull fashion into a

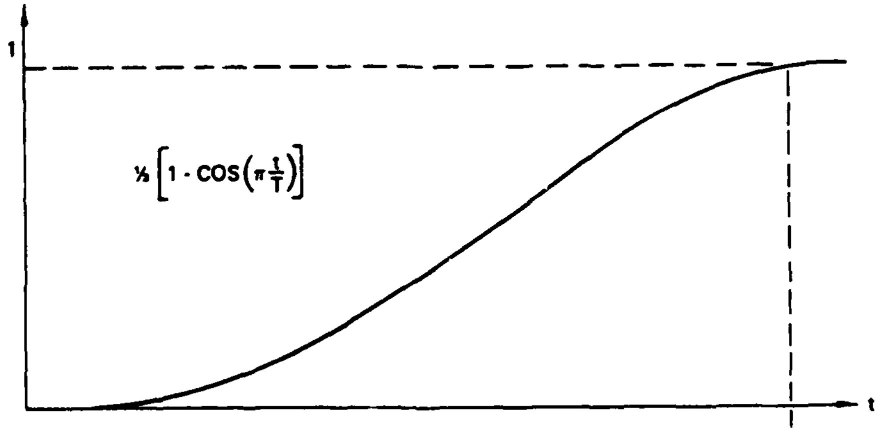


Figure 4.4-2. Approximate Step Response of Practical Receiver Filter

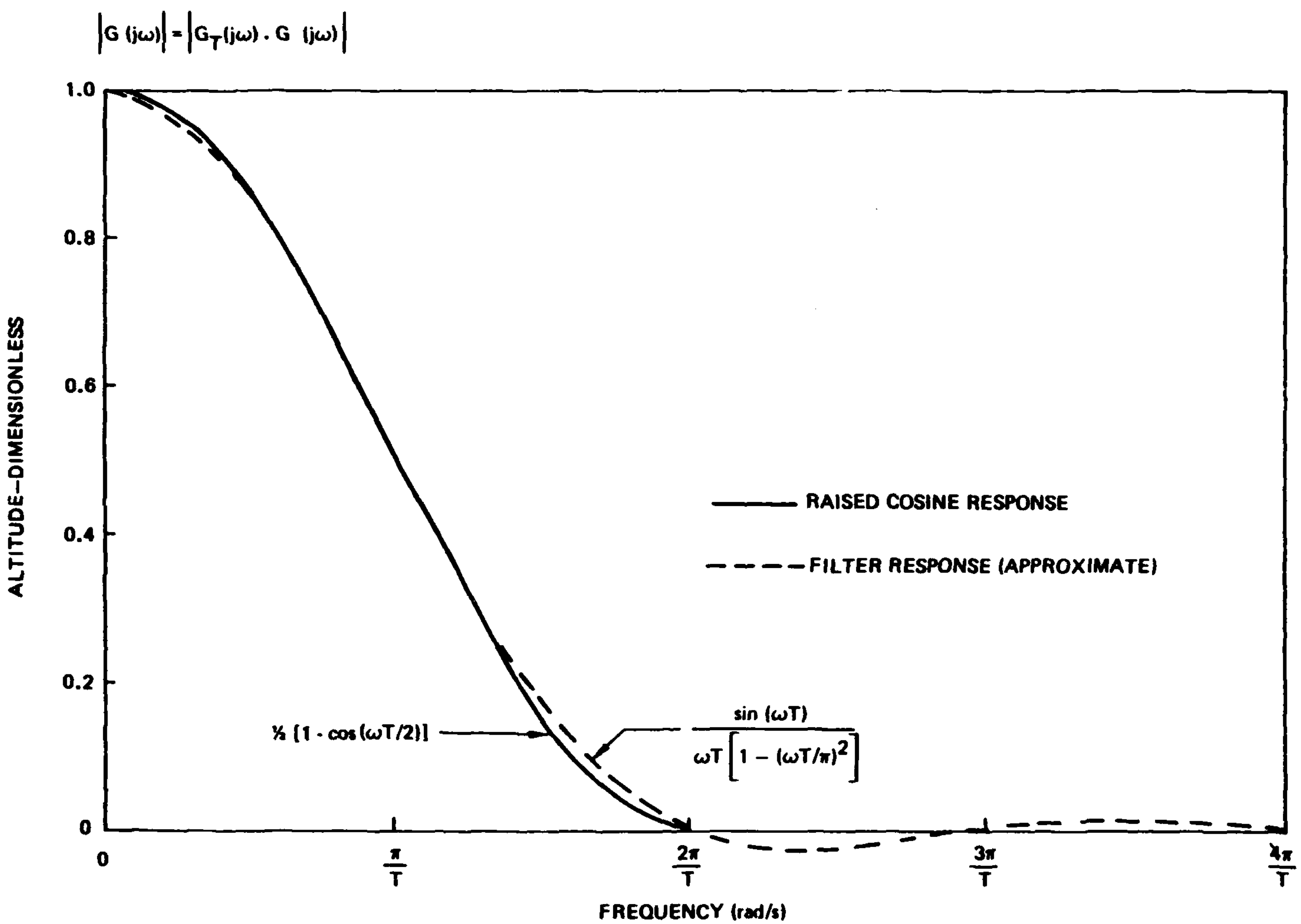


Figure 4.4-3. Desired and Approximate Filter Response

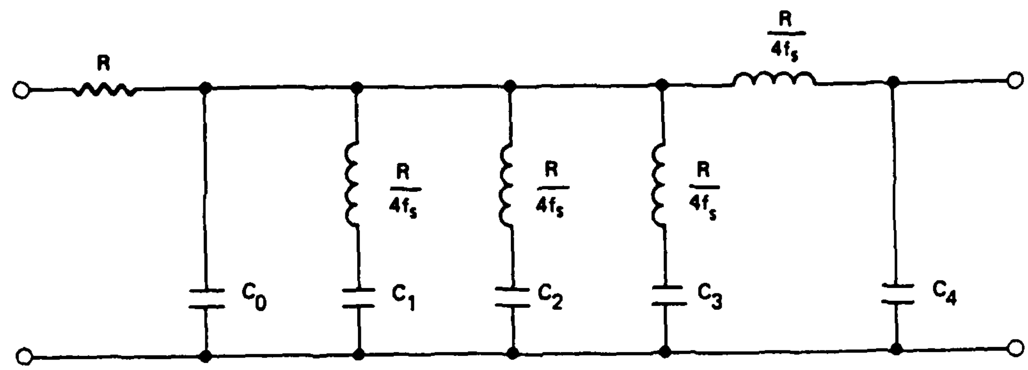


Figure 4.4-4. Schematic Diagram of Synthesized Filter

center-tapped winding on the coupling transformer. The selection of current or voltage mode drivers for implementation of the transmitter does not appear to be a significant factor affecting the performance capability or complexity of the hardware. The selection of driver type is more dependent on the designer's experience and prejudice.

Some major considerations for a design that meets the requirements of 1553B and the other performance criteria of the transmitter are --

a. Load impedance transformer characteristics
b. Output waveform (waveshape control)
c. Output signal symmetry
d. Power dissipation
e. Fault conditions and shutdown control

<u>Load Impedance and Transformer Characteristics</u>

The implementaton of the transformer in the coupling network can have significant effect on the signals transferred to the bus and the transmitter performance. The turns ratio required for the two bus connections (direct- or transformer-coupled) are different by a factor of 1.41. The load

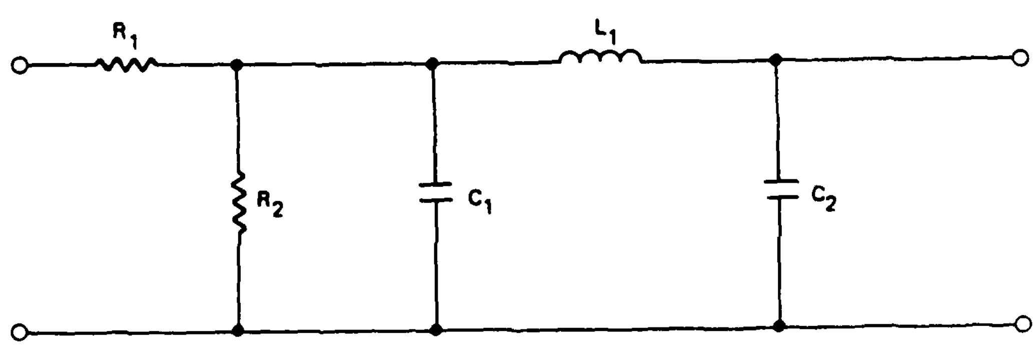


Figure 4.4-5. Schematic of Simplified Raised Cosine Filter

impedance reflected back to the transmitter in each case should be approximately the same and would be approximately 70 ohms if turns ratios of 1:1.41 and 1:1 are assumed for the transformer-coupled and direct-coupled connections, respectively. The actual transmitter load will depend on the turns ratio selected for the transformer. There are significant interrelated effects between the transmitter and transformer and the combination must be designed to meet the requirements specified in 1553B.

Output Waveform

A decision must be made early in the design phase to define the required transmitted waveshape. The 1553B standard specifies a range of rise and fall times with allowed zero crossing deviations and droop and overshoot limits. The standard allows the signal to vary from a trapezoidal waveshape with limited rise and fall times to a sine wave. This variety of signal types has been implemented on various data bus systems in the past. The generation of a sine wave at the terminal connection to the bus requires extensive filtering and linear driver resulting in increased complexity and cost and a significant penalty in power dissipation and weight. This problem is discussed in some detail in section 4.2.1.2 under "Output Waveform." The conclusion is that special filtering at the transmitter adds complexity and cost and is not required for most applications. The detailed characteristics of the output waveform should be selected prior to start of transmitter design.

Symmetry

The symmetry of the positive and negative output signal in time and drive is specified by 1553B. It defines the maximum signal level (tail) at a point 2.5 us after the midbit zero crossing of the parity bit of the last word in a message transmitted by the terminal under test. A well-designed driver is essential to reduce the detrimental effects caused by the unbalance. The test conditions are defined so that the messages contain the maximum number of words with various bit patterns selected for worst-case conditions. The transmitter designer must ensure balanced drive while the circuits are exposed to the variations of temperature and long-term aging effects. This requirement also implies that the simpler the transmitter and the lower the power, the easier it is to control the balance at the output. This leads to another reason for using the simple trapezoid instead of the filtered linear waveform.

Power Dissipation

The power dissipation of the transmitter drivers should be minimized for improved reliability. The efficiency of a well-designed trapezoid waveform driver can be significantly higher than the sine wave driver. The penalties of increased size, weight, and cost, and the decreased reliability are significant factors to consider when making the waveform decision.

Fault Conditions and Shutdown Control

The transmitter designer should consider the effects of failure modes on terminal and overall system operation. It is highly desirable for the transmitter to turn "off" automatically when a failure occurs. This is not always possible. A timeout feature is specified in 1553B to preclude a

signal transmission time greater than 800 us. The outputs of the drivers are monitored and timed using either analog or digital timers. The approach indicated in figure 4.4-1 uses a digital counter and system clock for time measurement. Shutdown is accomplished at the driver with a minimum of intervening circuit elements. Additional controls for transmitter enable and timeout reset are included.

4.4.4 Packaging Considerations

The transceiver circuit components are not compatible with LSI packaging techniques and are usually implemented with discrete transistors, resistors, capacitors, diodes, inductors, and transformers. Certain linear integrated circuit amplifiers and comparators are used in the receiver.

The densest packaging technique available for components of this type is hybrid circuits. All components except the transformer, inductors, and large capacitors can be included in hybrid circuit packages. A number of hybrid circuit receivers, transmitters, and combinations have been built. Some of these devices are available from various manufacturers as standard product items to meet various versions of 1553 and derivatives of 1553 for specific programs where unique interface signal conditions have been specified.

Some typical examples of hybrid circuit devices that can be incorporated in transceivers are listed below with a brief description.

a. A hybrid circuit receiver includes most of the elements shown in the receiver section of figure 4.4-1. The circuit is packaged in a 1- by 1- by 0.150 in flatpack.

b. A hybrid circuit transmitter is designed to supply an output meeting the requirements of 1553A and a major aircraft interface specification requiring a filtered signal. The transmitter "on" time detector and timeout control circuit is not included. The transmitter is packaged in a 1.25 by 1.25 by 0.200 in flatpack.

c. A receiver and transmitter are included in a single hybrid circuit device. The receiver has the same capabilities indicated for the single function device listed above. The transmitter output waveform is a trapezoid meeting the requirements of 1553B. The combined circuits are packaged in a 1.25 by 1.25 by 0.170 in plug-in dual inline unit.

Other circuits for various applications have been designed. These are representative of the type of components readily available for use in transceiver designs.

4.5 TERMINAL DESIGN

This section will address the various aspects of multiplex terminal design as related to particular subsystem requirements. The different types of terminals defined by 1553 will be discussed as an introduction to the physical and functional partitioning requirements of practical interface hardware. Finally, examples of a broad selection of typical interface hardware designs will be presented to illustrate some of the various forms of interfaces that may be required by actual subsystem hardware.

4.5.1 Types of Terminals

The definitions section (3.10) of 1553B defines a terminal as "the electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate line replaceable units (LRUs) or be contained within the elements of the subsystem." A terminal is further categorized by 1553B as either a bus controller, bus monitor, or remote terminal. Each of these categories will be discussed and examples provided in sections 4.5.1.1, 4.5.1.2, and 4.5.1.3.

4.5.1.1 Bus Controller

MIL-STD-1553B, section 3.11, defines a bus controller as "the terminal assigned the task of initiating information transfers on the data bus." Notice that the definition does not necessarily depend on the physical design of the terminal but is determined by the assigned task of bus control. This implies that a terminal may have the capability of performing other functions, but during the time when it is assigned the task of bus control it is by definition a bus controller. This will indeed prove to be the case for the design examples in the following sections. Figure 4.5-1 shows the generalized terminal functional elements that apply to a bus controller.

4.5.1.2 Bus Monitor

A bus monitor is defined by section 3.12 of 1553B as "the terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time." A bus monitor, therefore, is unique in that it performs no transactions on the bus. It not only does not initiate information transfers as a bus controller, it is incapable of any response on the bus, including status response. In fact, the bus monitor does not require a transmitter so it may be a "receive-only" terminal. The monitor function is a fully passive one of stripping selected data from the bus without in any way disturbing normal bus transactions. A bus monitor may or may not be assigned a unique address but must be capable of receiving data addressed to any or all other terminal(s) that it is assigned to monitor.

Note again that the definition applies to the assigned task and not necessarily to hardware configuration. A terminal may act as a monitor during some mission or flight phases and as a bus controller during others if it is capable of performing either function. Figure 4.5-2 shows the generalized terminal functional elements that apply to a bus monitor.

4.5.1.3 Remote Terminal

A remote terminal is defined by 1553B, section 3.13, as "all terminals not operating as the bus controller or as a bus monitor." This means that an RT cannot initiate information transfers on the bus as a bus controller and does not perform as a bus monitor. It must respond to commands issued by the bus controller in a normal command/response manner. An RT is identified by a unique address that allows the bus controller to direct specific information to it. Figure 4.5-3 shows the generalized terminal functional elements that apply to a remote terminal.

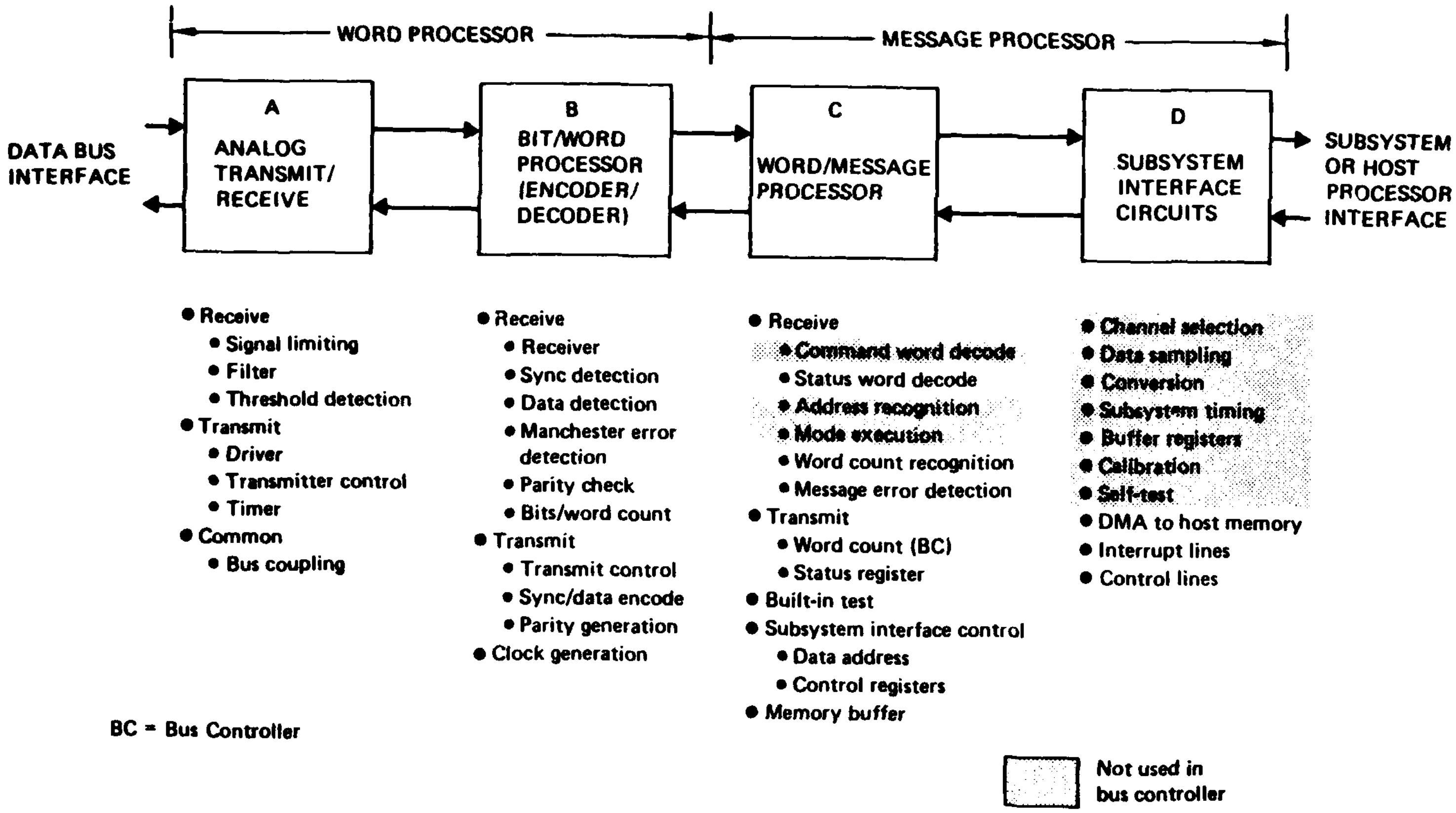


Figure 4.5-1. Bus Controller Functional Elements

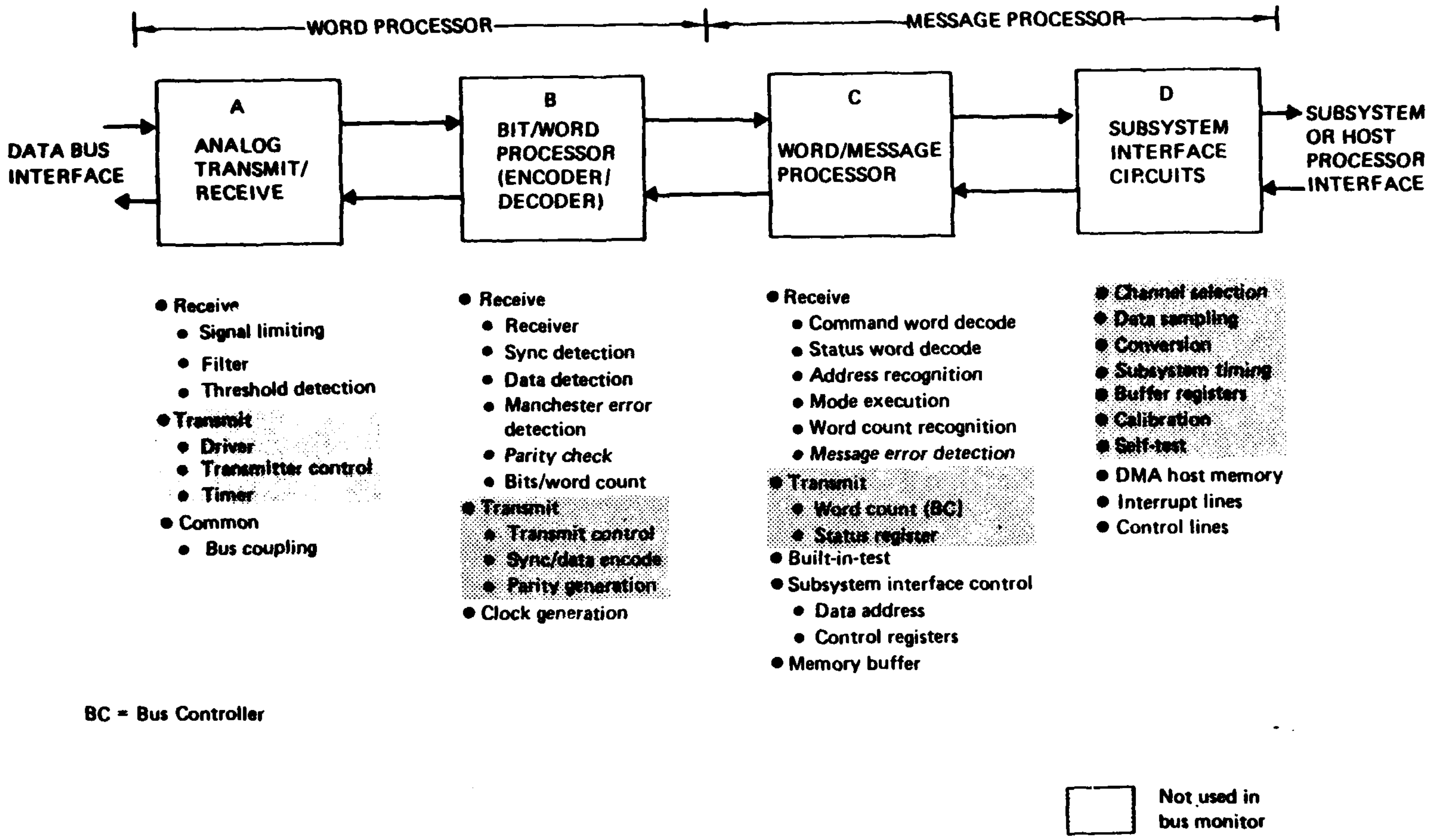


Figure 4.5-2. Bus Monitor Functional Elements

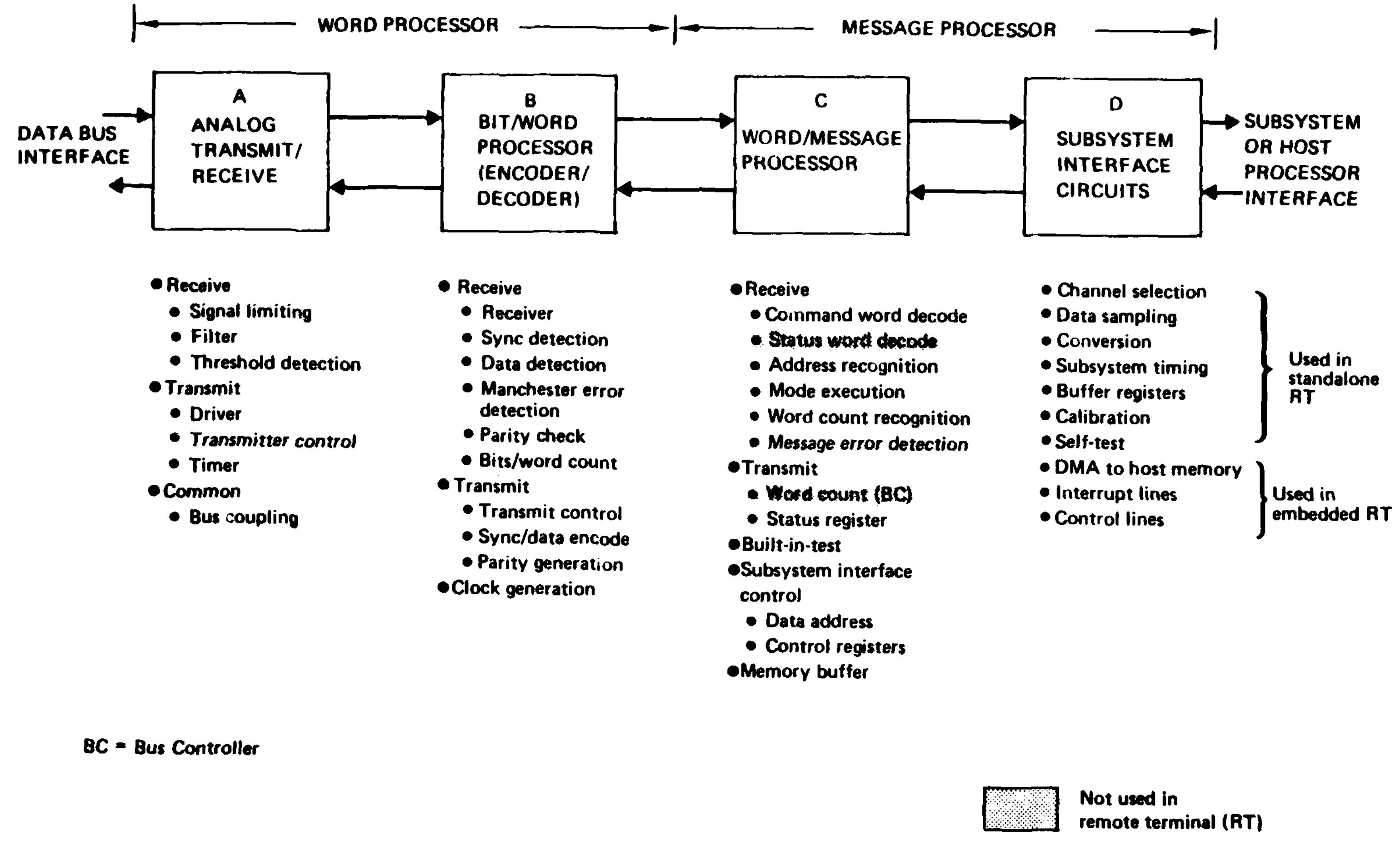


Figure 4.5-3. Remote Terminal Functional Elements

Once more it must be emphasized that the definition of an RT is one of function, not form. Any active terminal that is not performing bus control or monitor functions on a given bus at a given time is at that time by definition a remote terminal. It is a quite common practice in practical systems to provide bus control capability as well as RT functions within a terminal. This allows RT and bus controller roles to be passed about within the system as may be required for redundancy or specialized mission phase requirements. Also it is not unusual for a terminal with standby bus control capability to perform bus monitor functions while in a standby control mode. This method allows a sufficiently "smart" terminal to assume bus control if a set of predetermined bus transmission defects is detected.

A clear concept of the functional nature of these 1553 terminal definitions is essential to the understanding of the terminal partitioning and hardware example sections that follow.

4.5.2 Physical and Functional Partitioning

The functional elements of a generalized terminal were discussed in section 4.1.2 in which the terminal was divided into two major sections, the word processor and the message processor. These major sections were then subdivided into smaller subelements that may be present in a terminal. Figure 4.1-1 illustrated this breakdown. The elements depicted in figure 4.1-1 represent those functions that would allow a terminal to serve as either a bus controller, a bus monitor, or an RT. As shown in the preceding section, all these functional elements may not be necessary in a specialized terminal. For example, the transmitter function would not be necessary if a terminal were designed solely as a bus monitor. The fact that certain of these elements are required for the bus monitor function and a different set of elements may be required for the bus controller or RT functions is a basic functional concept that should be readily understood from the foregoing discussion of 1553 definitions.

A much more subtle partition, however, is suggested by the statement in the 1553B terminal definition that states "terminals may exist as separate line replaceable units (LRUs) or be contained within the elements of a subsystem." When a terminal exists as a separate LRU, there is no ambiguity as to where the terminal ends and the subsystem begins. All terminal functions are contained within the LRU (i.e., the terminal function stops at the physical partition). This terminal-subsystem line becomes more ambiguous, however, when the terminal is "contained within the elements of a subsystem." This problem is a result primarily of the fact that when a terminal is embedded within a subsystem, there may no longer be a clear-cut physical partition between the terminal and the subsystem. For example, if a terminal is embedded within a general-purpose computer, basic word processor functions (refer to fig. 4.1-1) may be provided by a clearly defined separate circuit card within the host computer, and message processing functions such as command decoding or address recognition may not exist as hardware at all but may be a software function of the subsystem. However, since by 1553 definition a terminal must "interface the data bus with the subsystem" and since this function is not complete until all required functions for a terminal are performed, part of the host software becomes by definition a part of the terminal and the physical partition becomes inaccessible. This is a very common situation in practical multiplex hardware. Whether the bus interface is a standard product or a

custom design, some of the terminal functions almost always reside within the user subsystem. In order to discuss practical hardware examples while maintaining strict adherence to 1553 definitions, therefore, it is necessary to establish some means of clearly defining the physical interface between the hardware and the subsystem as opposed to the functional interface, which may be inaccessible.

Because it has been established that a generalized terminal requires all the functional elements depicted in figure 4.1-1, regardless of whether they reside in the interface hardware or the subsystem, this figure will serve as a focal point for the discussion of practical hardware designs. By using figure 4.1-1 in this manner, a physical interface may be clearly shown that separates the functional elements contained within the interface hardware from that contained within the subsystem. Thus, the examples that follow will be referred to as interface hardware rather than terminals.

4.5.3 Typical Interface Hardware

This section presents five different examples of existing multiplex interface hardware designs using the physical and functional partitioning ground rules and definitions established in the preceding sections. The examples were chosen to provide a wide variety of hardware types, from the highly specialized to the extremely flexible. The three types of terminal functions discussed in section 4.5.1 are all represented. Some examples include the capability of performing all three terminal functions and others are restricted by design to a single function. The examples chosen, together with their functional capabilities, are listed in table 4.5-1.

For each example discussed, the generalized terminal functional element diagram (fig. 4.1-1) will serve as a focal point. This figure will be reproduced in each section with the appropriate functions highlighted and the interface hardware physical partition shown.

4.5.3.1 Bus Interface Unit

As previously noted, the analog transmit-receive portion of the remote terminal is usually implemented with discrete or hybrid circuits because of the type of components required (see sec. 4.1.2.1). Likewise, the system interface portion may contain analog and/or high-power driver circuits and is usually a direct function of the requirements of the particular user

Table 4.5-1. Interface Hardware Examples

Section	Example	Functional capability		
		Controller	RT	Monitor
4.5.3.1	Bus Interface Unit Chip Set (Harris)	X	X	
4.5.3.2	Standalone RT (B-52—Boeing)		X	
4.5.3.3	Multiplex Remote Terminal Unit (AAH—Sperry)	X	X	X
4.5.3.4	Flexible MUX Interface (SCI)	X	X	X
4.5.3.5	RT Embedded in Subsystem (Hughes LSI)		X	

subsystem. The encoder-decoder and word-message processor section (B and C of fig. 4.1-1), however, are common to all RTs and readily lend themselves to LSI implementation. Functions provided by the Harris bus interface unit (BIU) are shown in figure 4.5-4.

The BIU is an LSI approach to the implementation of the interface between the host electronics and the 1553B Manchester data bus. A two-chip approach was used because of the complexity of the system. One chip, BIU 1, can act as a standalone unit for use in a less complex system such as a remote terminal (RT). Where an additional capability is needed, the second chip, BIU 2, is used to enhance the operational capability of BIU 1. Figures 4.5-5 through 4.5-8 show basic block diagrams and functional pinouts of the two chips. In the discussion that follows, the two-chip set will be discussed as a single hardware interface.

This section describes the functions of the BIU primarily from the perspective of a bus controller and discusses the operation briefly as a bus interface for a processor acting as a remote terminal.

A key point to keep in mind is that software must control and interpret the entire interface as described here. Error determination and handling must be based on the specific BIU interface that exists for a given piece of hardware. Each BIU operates as an independent I/O channel in contrast to some BIUs that share common receivers and transmitters. In this implementation, assume that one BIU controls bus A and the other BIU controls bus B. Each has its own set of registers, which are described below.

4.5.3.1.1 Stored Program Instruction Interface

During initialization, the BIU is given an address for its instruction address register (IAR), which points the BIU to its BIU instructions. Instructions are arranged in pairs, which are stored sequentially in memory. The format for these instructions is given in table 4.5-2. The BIU is also given a base address register (BAR), which is 10 bits long.

Using this and the instruction words, the BIU can develop the address into the pointer block and find the data buffer. The first of a pair of direct memory access (DMA) sequences occurs, the first instruction word is acquired, and the address register is incremented. The BIU initiates a second DMA cycle and acquires the second instruction word. The IAR is incremented once again to prepare for the next fetch operation.

Once the two instruction words are acquired, the BIU can construct the command word(s). Referring to table 4.5-2, the BIU compares its terminal address (available from the PCR control word given to it when initialized by the host) with the device addresses in the instruction words. If the BIUs address is the same as the receive device address, then the command to be generated is a transmit command to an RT. If the BIUs address is the same as the transmit device address, then the command to be generated is a receive command to an RT. If the BIUs address does not agree with either device address, then an RT-to-RT pair of commands is to be generated. As part of built-in-test (BIT), the BIU checks to ensure that the receive device address is different from the transmit device address.

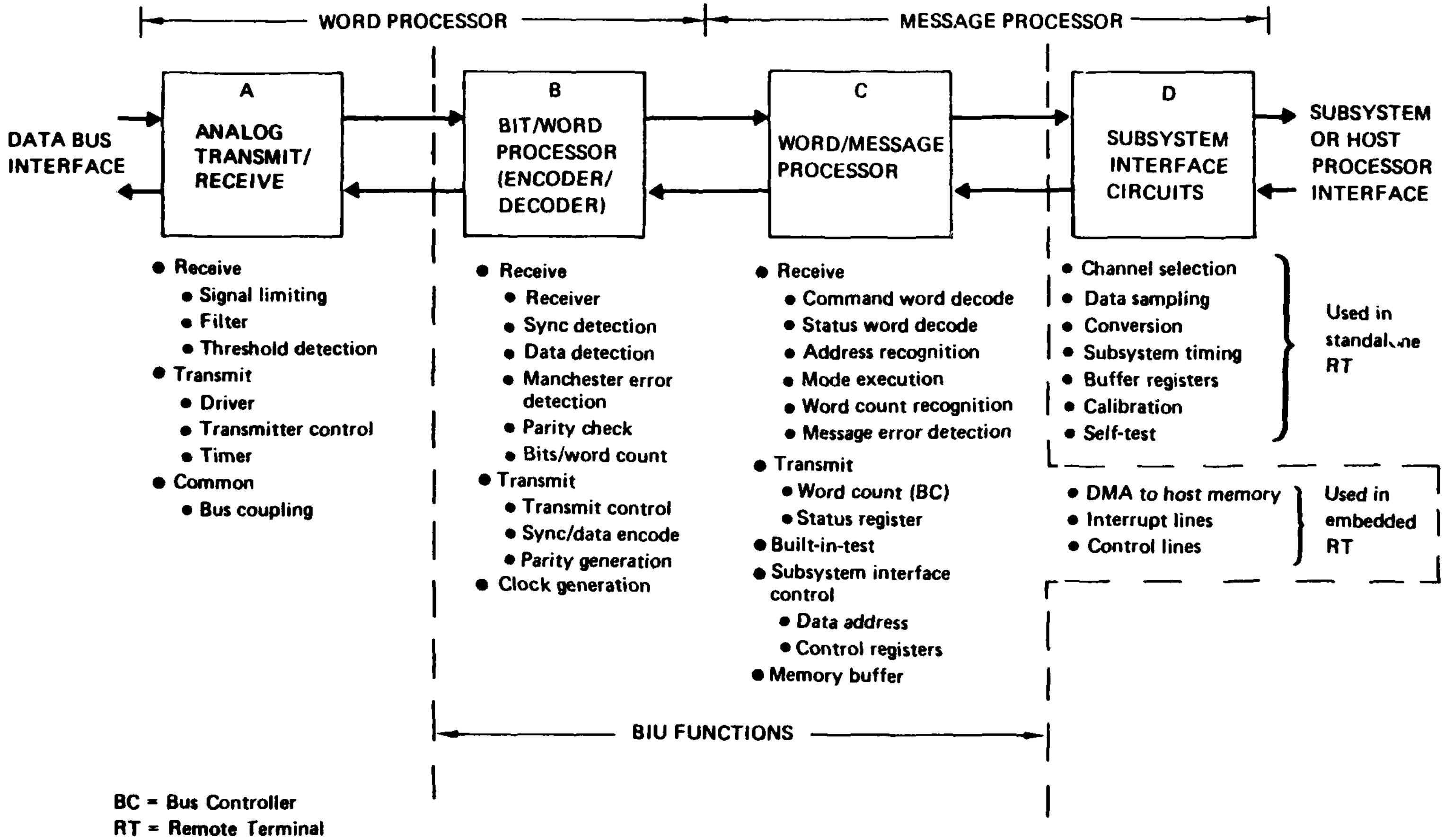


Figure 4.5-4. Bus Interface Unit (BIU) Functional Elements

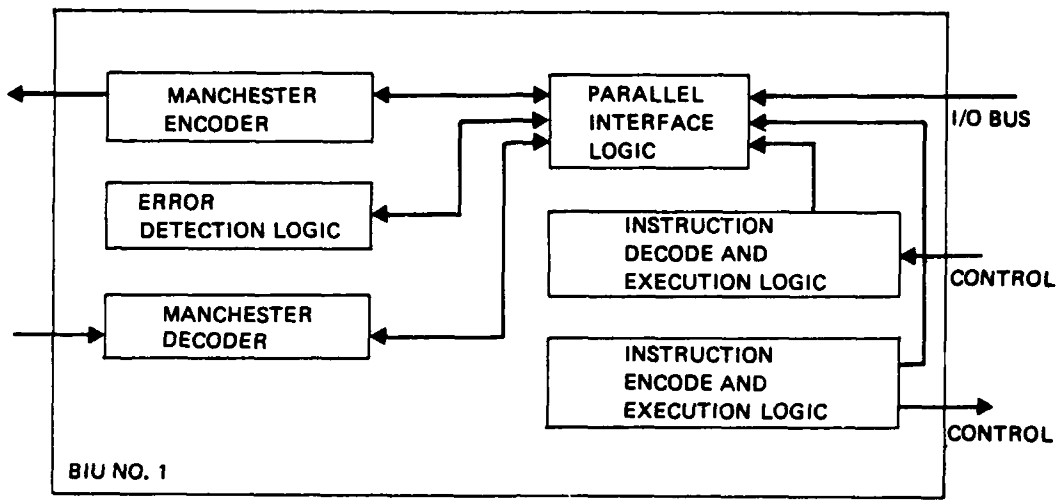


Figure 4.5-5. Bus Interface Unit No. 1 Basic Block Diagram

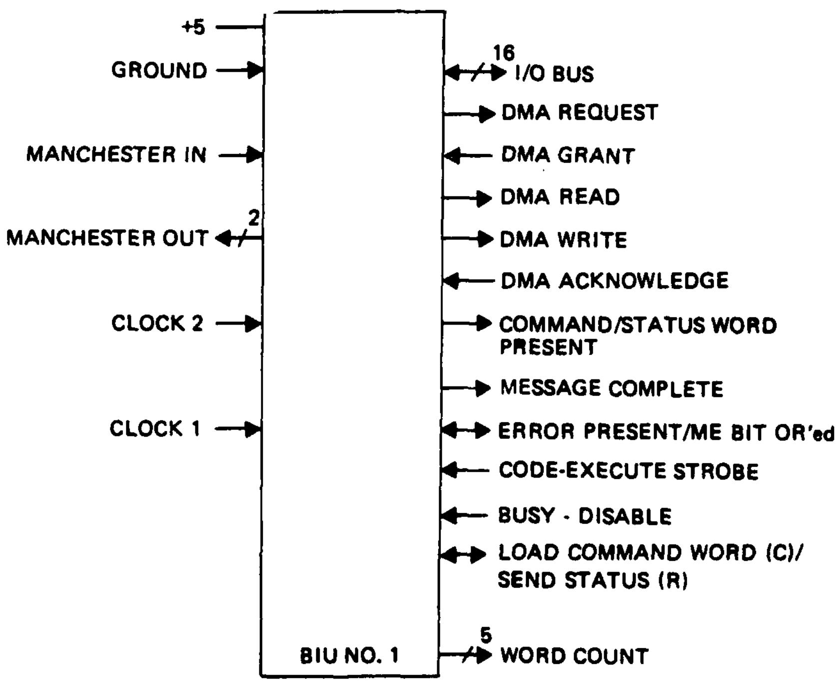


Figure 4.5-6. Bus Interface Unit No. 1 Functional Pinout

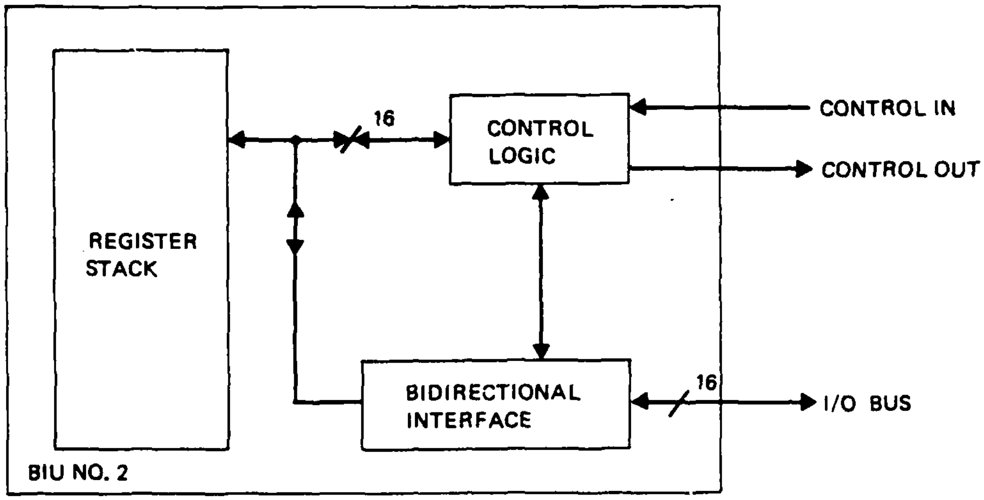


Figure 4.5-7. Bus Interface Unit No. 2 Basic Block Diagram

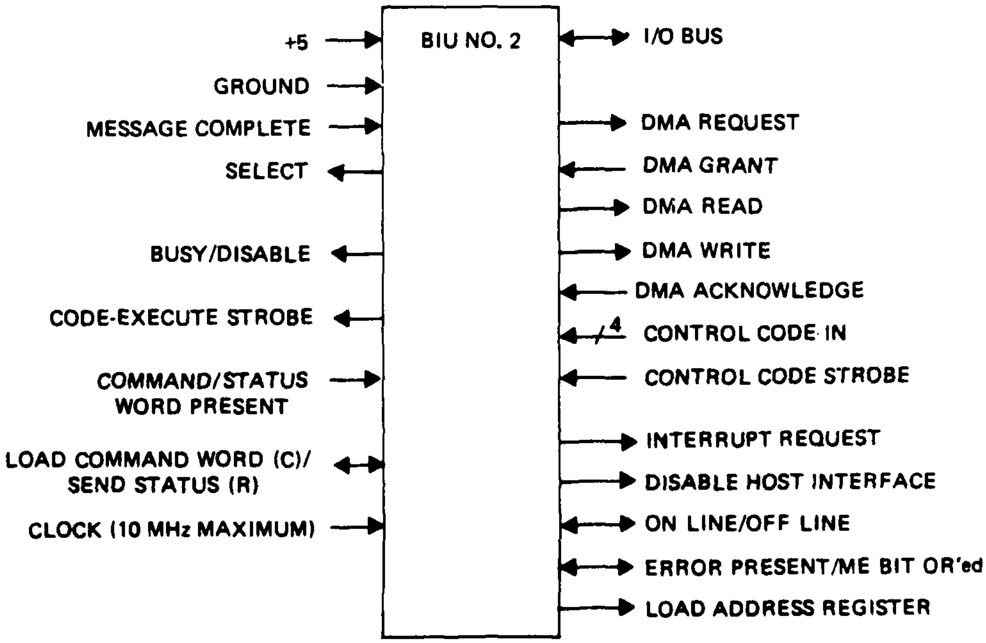


Figure 4.5-8. Bus Interface Unit No. 2 Functional Pinout

Table 4.5-2. BIU Instruction Format

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16 (LSB)
OP code		Retry		B	I	Receive device address					Receive subaddress/mode				
Word count/ mode code					S	Transmit device address					Transmit subaddress/mode				

IW1 bit designation

1-2 Normal OP codes
Bit 1-2
00 = Halt BIU
01 = Link (use second word as address of next two-word instruction)
10 = No operation (go to next two-word instruction)
11 = Normal bus operation

3-4 Indicates number of retries (0,1,2, or 3)

5 0 = Operation is performed on bus A
1 = Operation is performed on bus B

6 1 = Interrupt processor upon successful bus operation

7-11 Receive terminal addresses

12-16 00000, 11111 = mode command operation
00001 – – 11101 } Receive terminal subaddress
11110 = asynchronous message

IW2 bit designation

1-5 Word count or mode command code

6 Select: Select bit 0 = Select output 0; Select bit 1 = Select output 1

7-11 Transmit terminal addresses

12-16 00000, 11111 = mode command operation
00001 – – 11101 } Transmit terminal subaddress
11110 = asynchronous message

When an RT-to-RT set of command words is formed, no data buffer address is generated because the controller does not transfer data in that case unless the RT-to-RT data enable is set in the PCR (see fig. 4.5-9). When an RT is to receive data, an RT transmit command is generated and the BIU generates a corresponding data buffer address. As mentioned above, the BIU is given a BAR word (discussed later), and the BIU appends six bits (the T/R bit and subaddress bits) of the command word to the least significant end of the BAR word to form an address into a pointer table. The pointer, acquired from the table, points to the first address in the data buffer and is stored in the pointer register of the BIU. This first address is reserved for the tag word (time tag, word count, and data validity). The pointer address value is incremented and the value loaded into the BIUs address register for use when it executes its DMA transfers.

Once the data buffer address is set up, it is ready to transmit the command word. From that point, the BIU handles data transfer via its interface. If the message is an RT receive message, the data transfers by the BIU complete the message process; however, if the message is an RT transmit message that is received by the BIU, the data transfers to memory are followed by a final DMA transfer of the tag word into the first address of the data buffer.

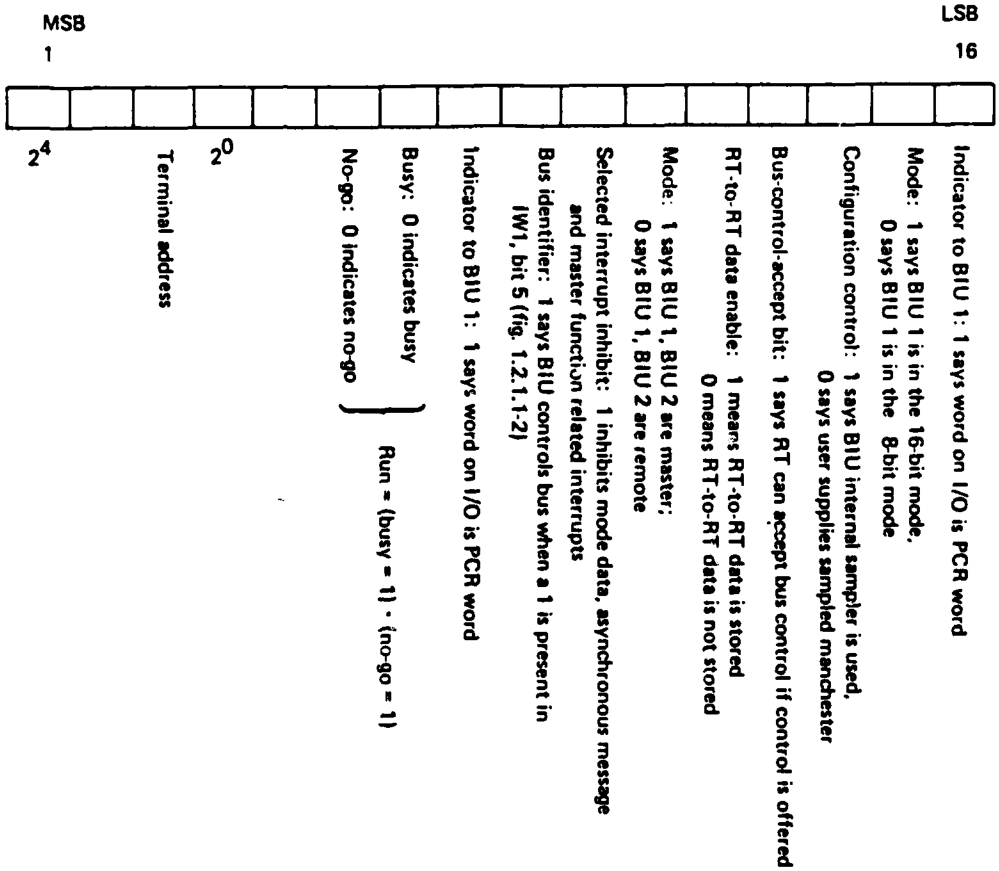


Figure 4.5-9. Processor Control Word Format

This process is summarized in table 4.5-3. The RT operation is summarized in table 4.5-4.

4.5.3.1.2 BIU Control Instruction Interface

Control instructions are used by the processor to initialize and to operate on the BIU. Control instructions occur either via programmed I/O or via memory mapped I/O. The control instructions in this example are treated as memory-mapped I/O instructions. These registers in the BIU are read or written much the same as the processor reads or writes its memory.

The control codes available to the host are:

0000	Load	Mode data register	(MDR)
0001	Load	Master function register (master)	(MFR)
0010	Load	Instruction address register	(IAR)
0011	Load	Base address register	(BAR)
0100	Load	Processor control register	(PCR) (both BIUs)
0101	Load	Status word data	(SWD)
0110	Load	Built-in-test register	(BIT)
0111	Halt	(graceful)	
1000	Output	MDR	
1001	Output	Receive status (master)/MFR (RT)	
1010	Output	IAR	
1011	Output	Transmit status (master)/BAR (RT)	
1100	Output	PCR	
1101	Output	Internal status register	(ISR)
1110	Output	BIT	
1111	Abort		

The Load Mode Data Register (MDR) instruction allows an RT or controller a place for storage of some special control word to be used in a later transfer. The controller BIUs processor might, for example, load bus designator information in the BIUs MDR and then direct the BIU to issue a mode command to an RT requiring it to shut down a selected transmitter. The controller would send the command word, followed by data from its MDR, to the RT.

The RT BIUs processor might, for example, load a service designator (or vector word) in its MDR using the Load MDR instruction. It would then set a service request (SR) flag using the Load Status Word instruction. Detecting the flag set in the status word, the controller BIU would generate an interrupt. The interrupted host, determining the interrupt cause, would request the vector word. When the transmit vector word mode command was received by the RT, the BIU would use the detection of the command as a reset signal for the SR and subsystem fail (SF) flag and return the 14-bit vector and SR and SF bits contained in the MDR. The resetting of the SR and SF flag has no effect on the contents of the MDR, which eliminates message errors affecting the eventual acquisition of the vector word through automatic retries.

The instruction, Load Master Function Register (MFR), allows a controller BIUs host processor to update the timing information (e.g., minor cycle number) used by the BIU when the BIU time-tags buffer data. The MFR data can also be transferred from the controller BIUs MFR to an RT's MFR by a

Table 4.5-3. Summary of Bus Controller BIU Message Processing

- BIU DMA's instruction words 1, 2 from host.

- BIU generates command word.

- BIU appends T/R and subaddress bits of command word to the least significant end of a 10-bit base address to form a 16-bit address into a pointer table:

LSB

BASE ADDRESS (10 BITS)	T/R	SUBADDRESS (5 BITS)

- BIU DMA's pointer-from-pointer table. The pointer is the location of the first address in the data buffer.

- Pointer is stored in the BIU's pointer register.

- BIU loads incremented value of pointer into external address register.

- BIU transfers command word.

- BIU handles data DMA's.

- In the case of a RT transmit message, the final data DMA by the BIU is followed by a DMA of the tag word into the first address of the buffer. The transferred tag word contains the minor cycle number, word count, and the data error bit:

MINOR CYCLE NUMBER (10 BITS)	WORD COUNT (5 BITS)	DATA ERROR

mode command (synchronize). If this mode command is broadcasted, all the RT BIUs are updated simultaneously.

The instructions affecting the IAR and the BAR, respectively, load a pointer to the set of BIU instructions and load the first 10 bits of the address of the message area.

The processor loads the BIU PCR (see fig. 4.5-9) to indicate to the BIU the BIU ID, bus ID (bus A or bus B controller), mode of operation (RT or controller), ability to accept bus control , GO/NO GO status, and busy status.

The Load Status Word Data instruction allows for the setting or resetting of the subsystem flag or the service request. These bits, at the disposal

Table 4.5-4. Summary of Remote Terminal BIU Message Processing

- BIU receives an RT transmit or an RT receive command.
- BIU determines that a command word is present.
- BIU determines if the command is an RT transmit or an RT receive command and begins data buffer address generation.
- BIU appends the T/R and subaddress bits of the command word to the least significant end of a 10-bit base address to form a 16-bit address into a pointer table.
- BIU DMA's pointer-from-pointer table. The pointer is the location of the first address in the data buffer.
- Pointer is stored in the BIU's pointer register.
- BIU loads incremented value of pointer into external address register.
- BIU handles data DMA's.
- In the case of an RT receive message, the final data DMA is followed by a DMA of the tag word into the first address of the data buffer.

of the RT processor, allow for additional (asynchronous) service from the controller beyond the periodic commands.

The Load Built-In-Test Register instruction allows the RT to report non-message-related failures (e.g., DMA handshake error) to the controller. Typically the controller, as part of a recovery procedure, would read the RT's BIT register.

4.5.3.1.3 Interrupt Interface

As mentioned previously, the BIU sets interrupts on various conditions:

a. Message errors
b. Status word exceptions
c. Certain mode commands
d. Program requirements

Interrupt generation reflects one of several facts:

a. The BIU has encountered a Manchester bus data transfer problem and error indications cannot be overcome without host intervention.

b. The BIU has been initialized because of a power dropout or startup and needs to be set up by the host.

c. The BIU has finished the bus-oriented tasks required of it by the BIU program in host memory and the program required host notification.

d. The host decides to intervene in BIU operation and commands the BIU to halt the operation gracefully.

As the word itself indicates, an interrupt is a break in an ongoing operational scenario, and when such a break occurs some trace of what happened must be recorded. Assume that each BIU possesses its own

interrupts so that the BIU that is interrupted can be identified. The possible reasons for interrupt generation are recorded by the BIU in the BIUs internal status register (ISR) (fig. 4.5-10) and built-in-test (BIT) register (fig. 4.5-11). Both these registers are available to the BIU host. These interrupts can be viewed from the perspective of the master controller or of the remote terminal. Both aspects are covered in the following paragraphs.

Interrupts Generated in the Master Controller

Message errors detected by the BIU include--

a. Manchester biphase errors
b. Word parity errors
c. No response
d. Message too short
e. Message too long

The BIU may diagnose a message error caused by failure of the preceding criteria. For example, an error could have occurred so that the sync detec-

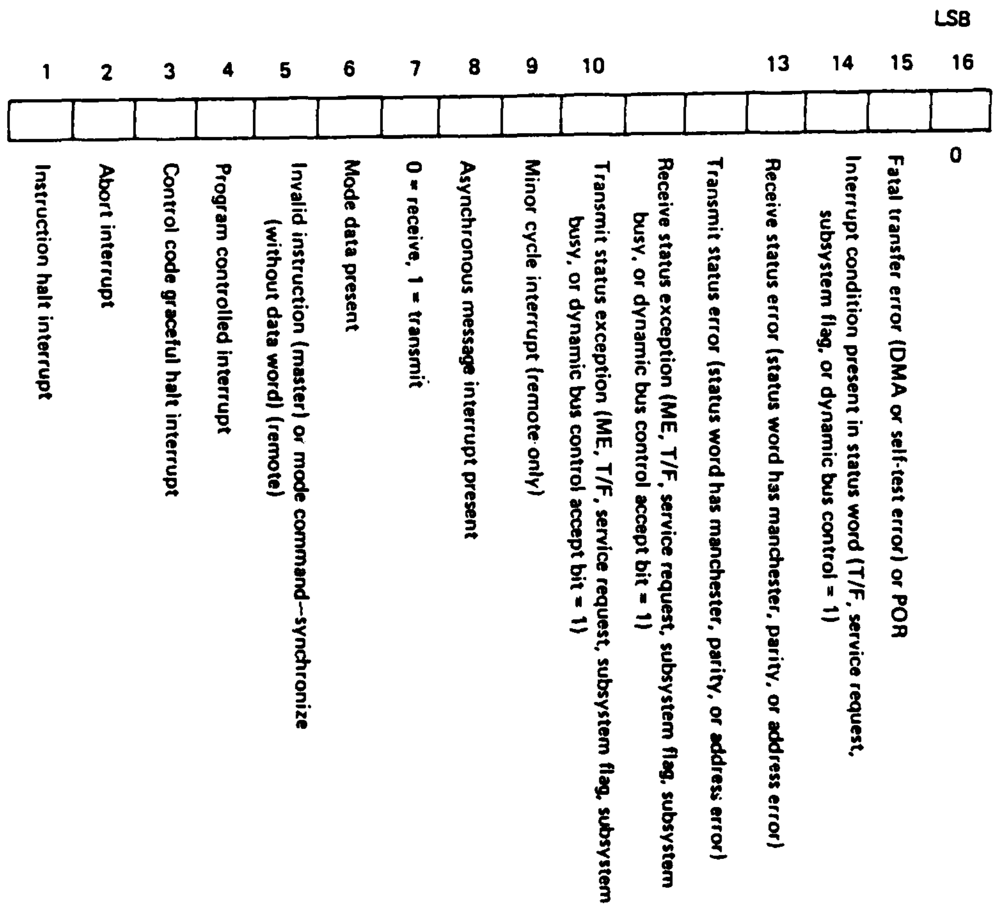


Figure 4.5-10. Internal Status Register

tion circuitry failed to validate the last data word of a given message because of distortion. Because the BIU did not detect the last word, it registers an error in message since the message was too short. The BIU has automatic retry capability and can be programmed for up to three additional message communication attempts, so hard failures tend to be separated from random error occurrences suggested by this example. In the case of a hard failure, the BIU will exhaust the retry attempt(s) and still find that a message error is present. When the BIU is involved in a message sequence, the BIU saves any indication of error-present until the message is completed. At the end of the message execution, the error present flag prompts the BIU to transfer the error word data (representing specific message errors, power-on reset, DMA error, and the loop-test error) into bits 4 through 16 of its own BIT register. After the transfer, the BIU tests for the presence of power-on reset, DMA errors, or loop-test errors. If power-on reset, DMA error, or loop-test error has occurred or the retry count is zero and a message error occurred, the BIU will interrupt the host and stop automatic operation. If none of these are present and the word count is not zero, the BIU clears the BIT word and ISR word and executes the next message sequence. If the status word is valid but contains some exception (e.g., T/F, SR), the error present flag will interrupt the host without any retry attempt; however, two status word exceptions (ME and subsystem busy) can cause retry attempts. These status word exceptions indicate the receive command was not received by the terminal and that a retry could rectify the problem if allowed. So, like the error present flag, the two exceptional conditions found in the valid status word prompt the BIU to test the retry count and either execute another message sequence or generate an interrupt. Besides generating interrupts when message error or busy conditions prevent successful communications, the BIU, acting as a master controller, generates interrupts in response to other conditions described below. In these cases, the BIU always stops operation until the interrupt is serviced. Under four conditions, the BIU will generate an interrupt when programmed to do so. The first condition can occur when the instruction pair being executed by the controller detects a message error or status word exception. Under these

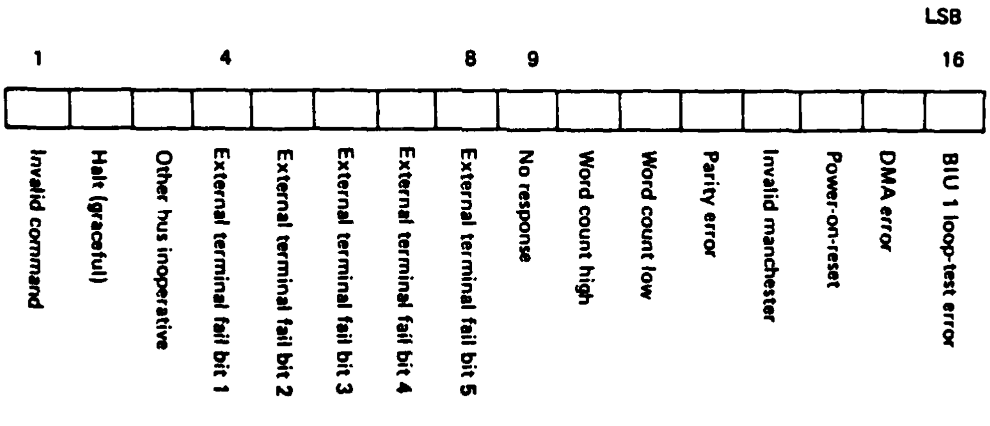


Figure 4.5-11. Bit Word Format

conditions, programmed message retries are attempted before generating the programmed interrupt. If the BIU is ultimately unsuccessful in accomplishing the instructed bus operation, the appropriate message-related conditions are recorded in bits 4 through 13 of the controller's BIT word, bit 4 of the ISR is set, and an interrupt is generated. It may be that during bus operation the BIU detects a loop-test error, DMA error, or power-on reset. In any of these cases, no retry is attempted, but the appropriate condition is recorded in bits 14 through 16 of the BIT word. Then, bit 4 of the ISR is set and an interrupt is generated. A second way of programming an interrupt is with the OP code of the instruction pair (see table 4.5-2). This can be set to require the BIU to halt. In this situation, the BIU decodes the requirement and executes it by setting bit 1 of the ISR and then generating the interrupt. A third method used to interrupt the host occurs when the BIU is given the control code 1111 by the host. This causes the BIU to stop all operations without regard to where it may be in its operating sequence. The BIU responds by setting bit 2 of the ISR and generating an interrupt. A fourth interrupt method is the graceful halt, which causes an interrupt based on the control code 0111 set by the host. The host can request the BIU halt operation, but in that case the BIU finishes its present operation, sets bit 3 in the ISR word, and interrupts the host. The graceful halt is executed identically to the program-controlled interrupt: only the indicator (in ISR bit 3 rather than the ISR bit 4) is different.

Interrupts are initiated by the BIU when the BIU discovers that an instruction pair contains the same device address in both instruction words. This check is made during command generation, and if this condition exists the BIU operations are halted without ever beginning data bus transmission, and bit 5 of the ISR is set. Bit 6 of the ISR is set and an interrupt is generated by the BIU when a mode command requiring mode data has been executed. If, while attempting to acquire the mode data, the controller detects a message error or status word exception, programmed message retries are attempted if allowed. If the bus operation is ultimately unsuccessful, the appropriate message-related conditions are recorded in bits 4 through 13 of the controller's BIT word, bit 4 of the ISR is set, and an interrupt is generated.

Bits 7 and 8 of the ISR are associated with the execution of an asychronous message. Bit 8 indicates the BIU participated in an asychronous message, and bit 7 indicates the BIU was the transmitter (bit 7 = 1) or the receiver (bit 7 = 0) of the message. The BIU processes these bits whenever it generates a command word with subaddress equal to 30. After executing the message associated with this command, the BIU generates an interrupt to the host. Treatment accorded message errors, specific status word exceptions, etc., is identical to that used for ISR bit 4 described in the preceding paragraphs.

Bit 14 of the ISR is set and an interrupt is generated when the status word contains an interrupt condition. Treatment accorded message errors, specific status word exceptions, etc., is identical to that used for ISR bit 4. The conditions in the returned status word that interrupt the controller include T/R, service request, subsystem flag, and dynamic bus control acceptance. It is assumed that automatic message retries would not be scheduled in sensitive cases (e.g., use of the dynamic bus control mode command).

Interrupts Generated in the Remote Terminal

In the preceding section ("Interrupts Generated in the Master Controller"), the text describes how message errors detected by the BIU are transferred into the BIUs BIT register. This same process occurs in the remote terminal mode. Once a transfer is made, the RT tests for the presence of power-on reset, DMA errors, or loop test errors. If any of these are present, the BIU will generate an interrupt. These are the only message-error-related failures that can cause interrupt generation in the RT.

In addition, the BIU as part of the RT configuration generates interrupts in response to other conditions described below.

Bit 2 of the ISR is set and an interrupt is generated when the abort command (control instruction 1111) is given to the BIU. The host can use this command to stop all operation without regard to where the BIU may be in its operation (transmitting or receiving).

Bit 5 of the ISR is set and an interrupt is generated when the RT BIU receives a valid message containing the synchronize mode command (without data word).

Bit 6 of the ISR is set and an interrupt is generated when the RT BIU receives any of the mode commands, 10000 through 11111 (except 10001), provided that the T/R bit of the mode command is a zero. Under such conditions mode data is waiting for the host in the BIUs MDR.

Bits 7 and 8 of the ISR are associated with the execution of an asynchronous message. Bit 8 indicates the BIU participated in an asychronous message. Bit 7 indicates the BIU was the transmitter (bit 7 = 1) or the receiver (bit 7 = 0) of the message. The BIU processes these bits whenever it receives a command word with subaddress equal to 30. After successfully executing the message associated with this command, the BIU generates an interrupt to the host.

Bit 9 of the ISR is set and an interrupt is generated when the synchronize mode command (with data word) is validly received. Upon reception of this command, minor cycle time information is waiting for the host in the BIUs MFR.

4.5.3.2 B-52 OAS Remote Terminal

The two card B-52 OAS RT developed by Boeing consists of a dual modem card to interface with two multiplex data buses and a handshaker card containing a 256-byte buffer memory. Except for initialization, the OAS RT operates independent of the user who interfaces with the handshaker card. Data words received and/or transmitted over the data bus are stored in or obtained from the buffer memory. Figure 4.5-12 shows the functional partitioning of the OAS RT. The modem card performs the word processor functions as shown. There are two independent modems on this card, one for each multiplex bus. The handshaker card provides the message processor functions of the RT.

A simplified schematic of the OAS RT is shown in figure 4.5-13. The output of the two independent modems is combined off the card and passed to the handshaker. When the modem receives a command word with its terminal

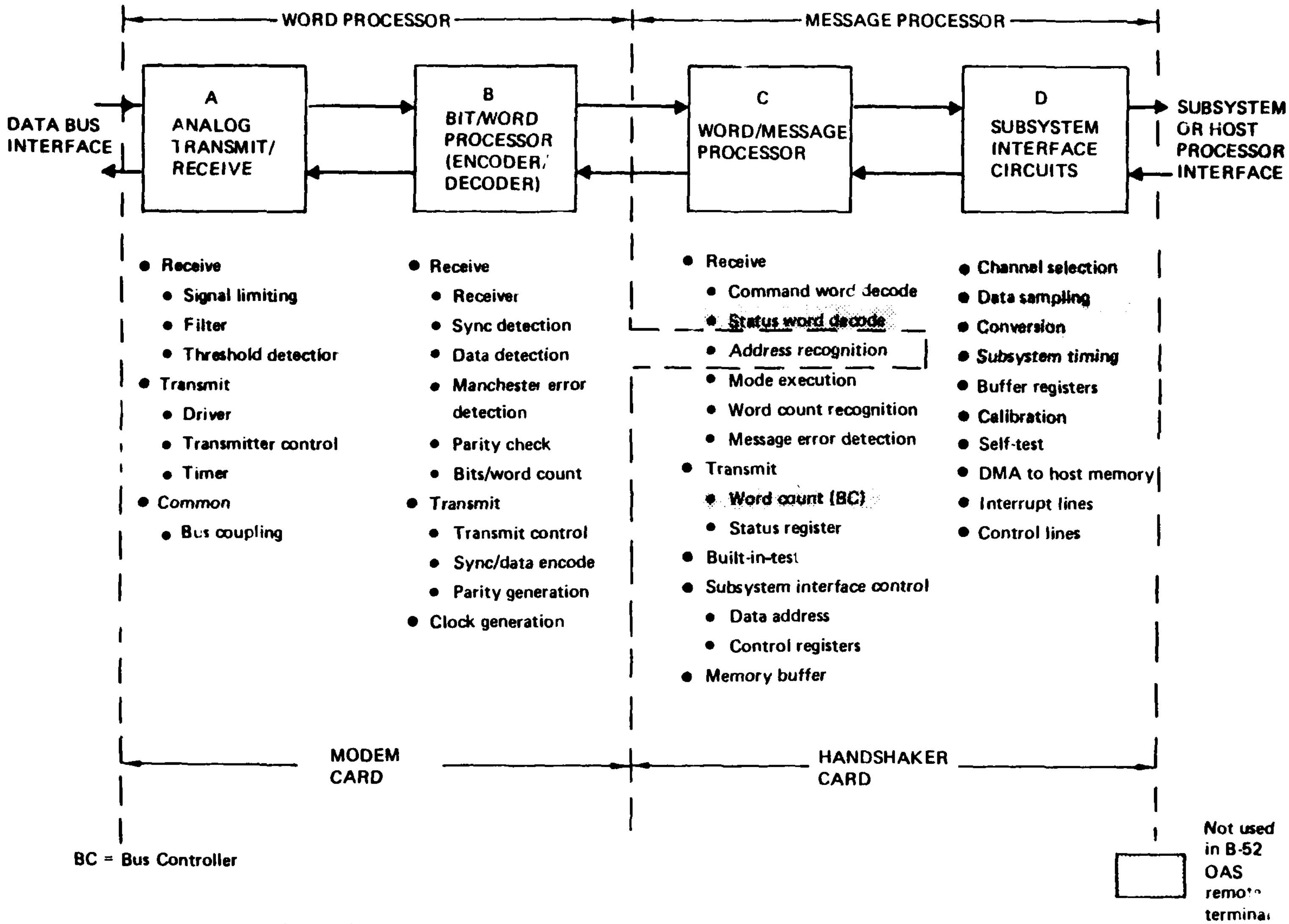


Figure 4.5-12. B-52 OAS Remote Terminal Functional Elements

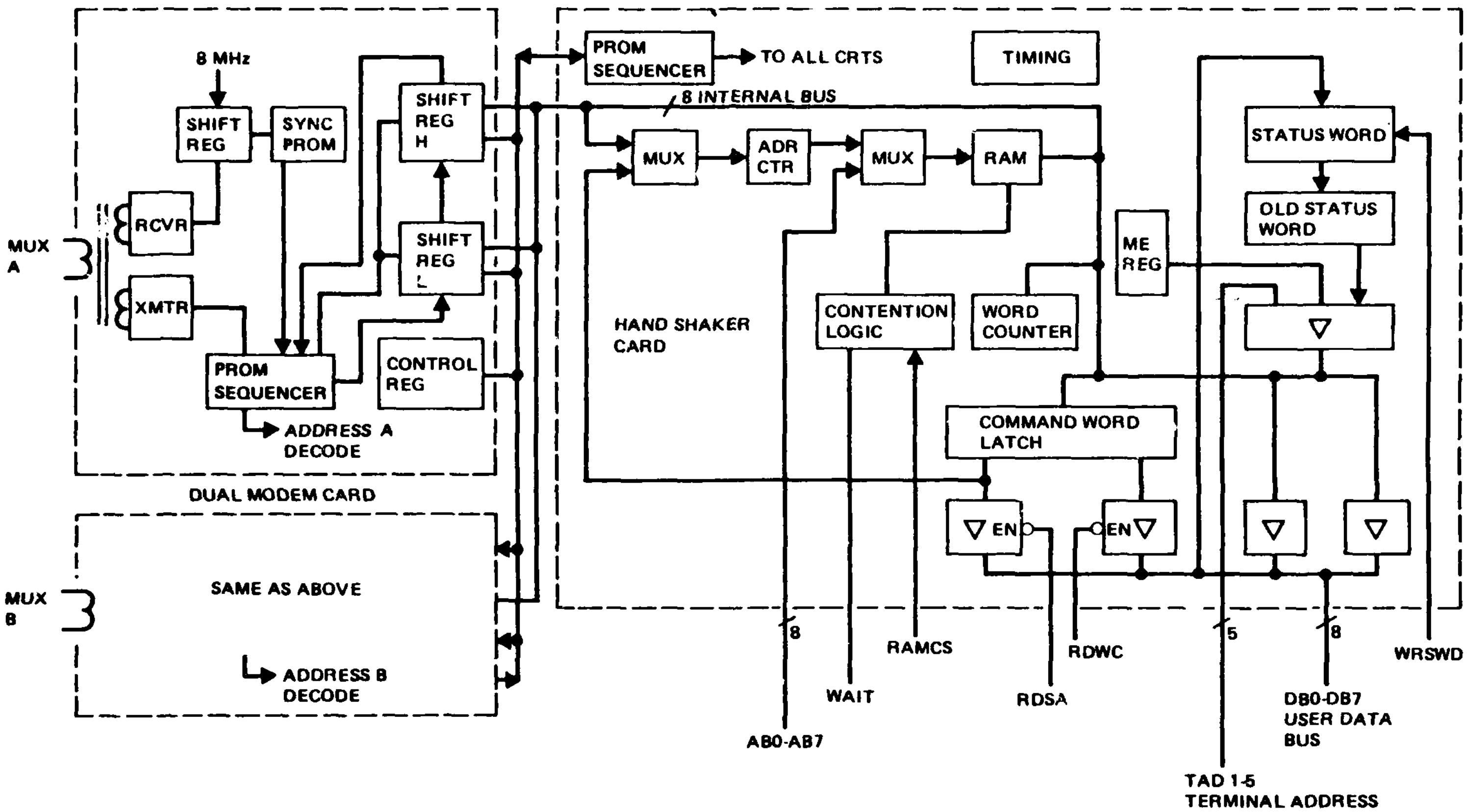


Figure 4.5-13. B-52 OAS Remote Terminal Simplified Schematic

address, it will signal the handshaker that there has been an address compare on that channel. The handshaker has control over the output enables of both modems and will only listen to the channel with the most recent address compare. In the receive mode the serial data from the multiplex bus is shifted into shift registers. It is read out a byte at a time by the handshaker. In the transmit mode the handshaker loads the shift register in parallel and the modem transmits it serially.

The modem detects and generates syncs, decodes the terminal address, counts bits within a word, checks for valid Manchester data, and detects and generates parity. The handshaker decodes the T/R bit, subaddress and word count from the command word and transfers the data words between its buffer memory and the modem card. During a modem transfer, the handshaker controls the internal bus and passes the data words between the modem and the memory at a location depending on the address generated by the mapping information derived from the given subaddress and T/R bit. Data are transferred over the internal bus a byte at a time, starting with the least significant byte of data. In addition to the data words in random access memory (RAM), i.e., the buffer memory, the user can read the most recent command word.

Before the OAS RT will operate, the subsystem must initialize the RT with mapping information. This is stored in the first 64 bytes of RAM. Whether the subsystem desires a nonmaskable interrupt after each valid transmission will depend on the state of the least significant bit (LSB) in the mapping information for that particular T/R bit subaddress combination. The desired status word (except for the terminal address and T/R bit) should then be placed on the subsystem bus. The transmit status word, transmitter enable, and transmitter disable mode codes are handled automatically within the handshaker. If any other mode code is received, the subsystem will be issued a nonmaskable interrupt. Transmitter enable and disable mode codes will not cause a nonmaskable interrupt but the transmit status word will if the LSB of RAM address is set.

4.5.3.2.1 Programmable Read Only Memory Sequencer

The modems and handshaker are both controlled by similar vertical programmable read-only memory (PROM) sequencers. The basic architecture of this sequencer is shown in figure 4.5-14. The signals from the output registers are used to control various functions within the modem and handshaker. The vertical sequencer is characterized by its serial nature. To lead each of the output registers shown in figure 4.5-14 and to perform a jump in the program would require four instructions of two cycles each. Control of the sequencer is governed by the instructions stored in PROM. All instructions consist of two adjacent bytes in PROM, which has a capacity of 256 instructions. The first byte is the OP code and is always in even address locations. The second byte of the instruction is data, which can be loaded into one of the output registers, into the program counter, or ignored (as in a NOP). The OP code is loaded into the instruction register using a 4 MHz clock. The OP code (i.e., the output of the instruction register) controls the routing of the data byte.

The modem and handshaker PROM sequencers have different basic capabilities and therefore they have different control requirements. To facilitate writing the sequencer firmware, a cross assembler was used so that the final

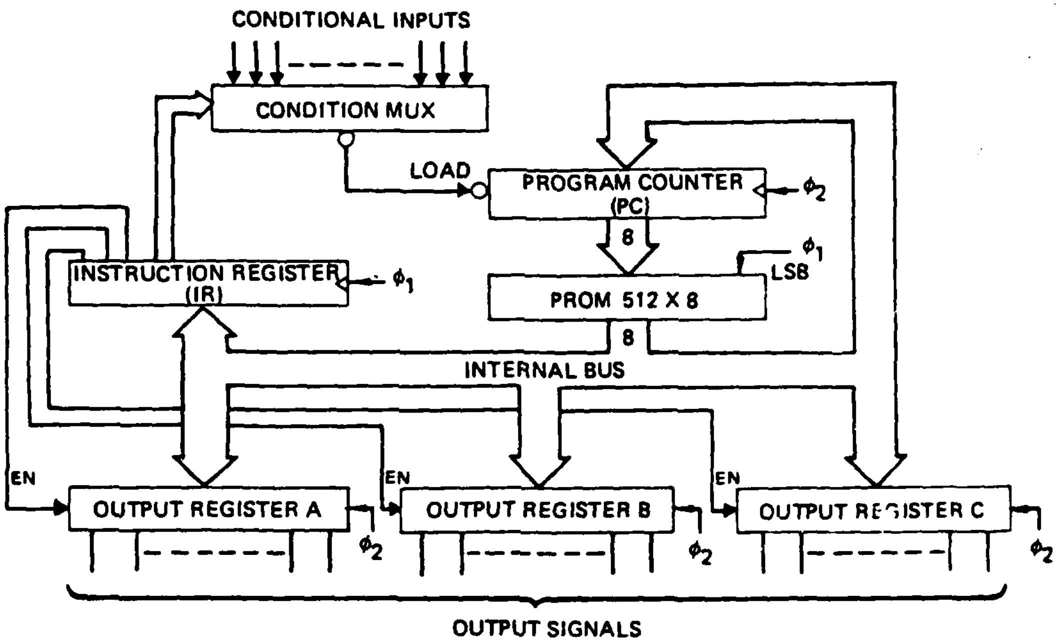


Figure 4.5-14. Vertical PROM Sequencer

object code could be automatically derived from a readable source listing containing mnemonics and comments.

4.5.3.2.2 Handshaker

The handshaker is the interface between the modem and the subsystem. It contains a 256-byte buffer memory (RAM) that all modem data words are transferred to or from. The subsystem has access to the contents of the buffer memory so that data words can be read or updated. The modem and handshaker communicate through several handshake signals.

There is some initialization required by the handshaker, but normally the OAS RT (modem and handshaker) can operate independently of the subsystems. The internal bus is eight bits wide and is used to pass data a byte at a time. It is isolated from the subsystem bus by tristate buffers. When the RT is idle (i.e., not handling a transmission on the data bus) the subsystem has immediate access to the RAM. During data bus transmissions, the handshaker will take control of the internal bus to prevent subsystem access to the RAM.

The handshaker decodes and stores the subaddress, T/R bit, and word count from the command word. It counts data words and transfers them between the modem and RAM. It also supplies the status word and handles the send status word and enable-disable alternative transmitter mode codes.

The handshaker is designed to interface with a dual-channel modem but does not interface with both channels simultaneously. The tristate outputs of the two modem channels are tied together and controlled by the handshaker,

which will enable the channel with the most recent address compare. Thus, handshaking on one channel will cease if there is an address compare on the other. Handshaker firmware is designed so that an interrupted message will terminate in a known orderly fashion.

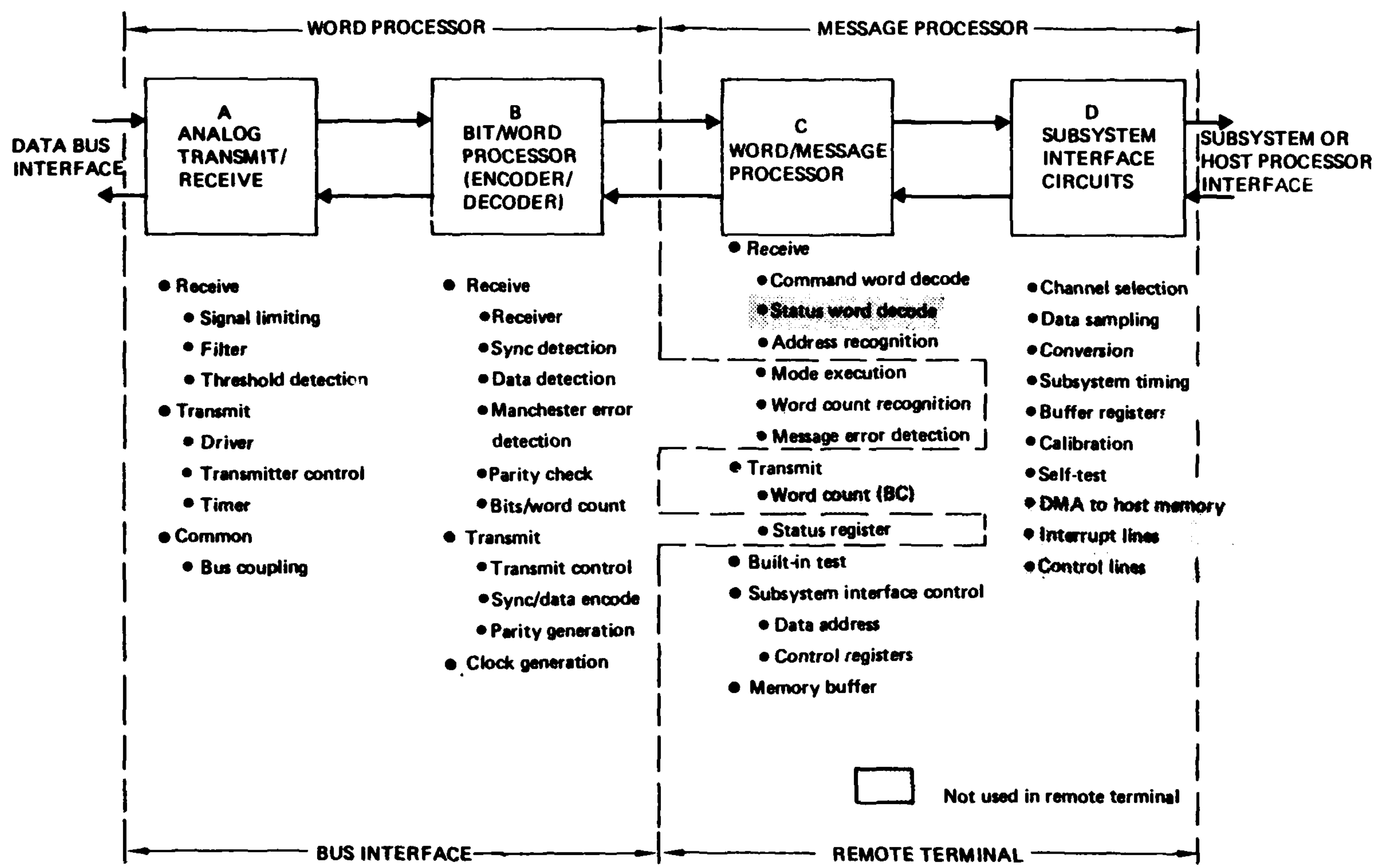
The 256-byte RAM is used to store data words for the modem. In addition, the first 64 bytes are dedicated to containing mapping information. Each T/R bit and subaddress combination in a command word can cause the associated data words to be transferred (mapped) to or from any of the 192 remaining bytes in the RAM. The location of the first byte of data is mapped into the address defined by the contents of the byte at the address made up of the T/R bit and subaddress. For example, assume a command word had a T/R bit = 1 and a subaddress = 00101, then the contents of 25H (00100101) contains the address of the first byte to be transmitted by the modem. If the contents of 25H is B4 and the contents of B4 and B5 is 58 and 7C, then a command word in the example will cause a data word of 7C58 to be transmitted. Note that the low byte of the data word is stored in the lower address.

4.5.3.3 Multiplex Remote Terminal Unit

The multiplex remote terminal unit (MRTU) can serve as a standalone remote terminal or simultaneously as an RT and a backup bus controller (BC). Figure 4.5-15 presents the functions performed in the MRTU developed by Sperry. The bus interface performs the analog transmit-receive and the bit-word processor functions. I/O control and I/O signal conditioning circuitry complete the RT and perform word and message processor and subsystem interface functions, respectively. Since the backup bus controller is colocated with an MRTU, there is a unique case where the backup bus controller has bus control and must communicate with the MRTU in which it is located (see fig. 4.5-16). What is important to note here is that this MRTU is treated the same as any other. The backup bus controller transmits its information on the MIL-STD-1553A bus via the transmit portion of the bus interface hardware within the MRTU while the receive portion of the bus interface hardware accepts and processes the information from the bus. There is no internal connection or communication between the backup bus controller and MRTU. Thus, each one treats the other as a completely external device, except for the common bus interface hardware.

4.5.3.3.1 Remote Terminal

Figure 4.5-17 presents a block diagram of the bus interface portion of the MRTU. The received Manchester data are filtered, synchronized to the internal clock, and monitored for a valid sync waveform. When a valid command sync is detected, the data are multiplexed to the decoder, which changes the data from Manchester to NRZ binary. The data output is provided in both parallel and serial form. The data are checked for odd parity and proper Manchester coding. Command words are checked for valid address by comparing the first five data bits with the terminal address. Data words are checked for valid sync timing. Appropriate bits of valid command words are transferred to the word counter, subaddress register, and the T/R register. The output of the word counter provides an address for the data words in a message. The outputs of the subaddress and T/R registers are used to select unique blocks of data to be transferred. The word counter and the data, in serial format, are transferred to the I/O controller. The heart of the I/O control is a sequence control PROM, which, along with the



BC = Bus Controller

Figure 4.5-15. YAH-64 Multiplex Remote Terminal Unit (MRTU)

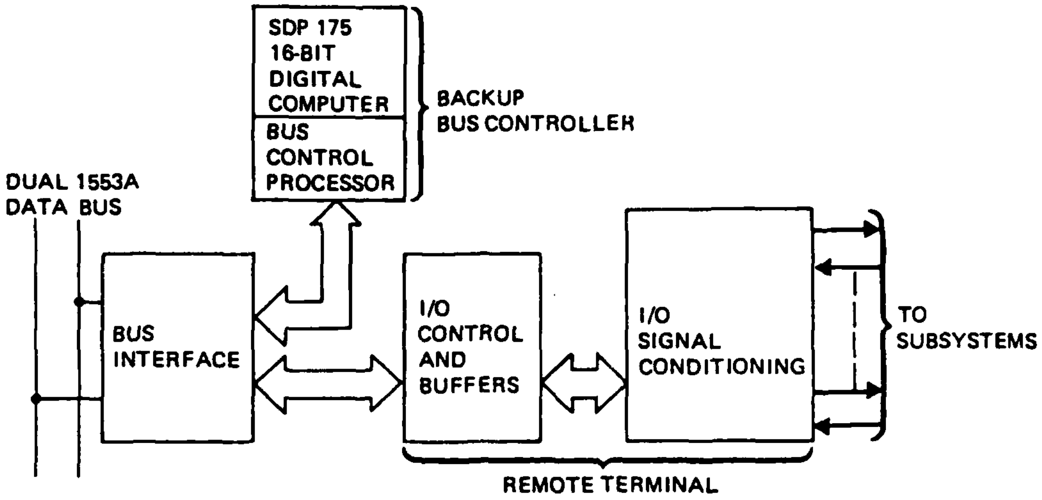


Figure 4.5-16. Remote Terminal With Colocated Backup Bus Controller

I/O timing and control hybrid circuits, provides a cyclic scan of all analog and discrete inputs and a cyclic update of all analog and discrete outputs.

The serial data received from the bus interface are stored in an output RAM memory in the I/O controller. Under sequence PROM control, the RAM data are cyclically presented to the analog and discrete output signal conditioners. The data to the analog outputs (except sync) are processed in a 12-bit D/A converter and transmitted on the output analog bus to the selected analog output signal conditioner. The discrete data and synchro data are transmitted directly from the RAM memory to the holding register in the selected discrete output or synchro output signal conditioner.

For the scanning of inputs, each input signal conditioner provides a differential analog signal to the I/O controller input multiplexer. The multiplexed signal is processed in a high-speed 12-bit A/D converter and stored in an input RAM memory for serial transmission to the bus interface. Each analog input is stored as a 16-bit word with the four LSBs set to zero. Each discrete input channel is stored as a single binary bit with 16 successive channels forming a data word.

Note that in the operation of transferring data between the MIL-STD-1553A bus and the I/O, no processor or bus controller is needed. This emphasizes the fact that an RT can operate with no internal bus controller and, conversely, a backup bus controller can be located inside an RT with complete independence.

4.5.3.3.2 Backup Bus Controller and Remote Terminal

The backup bus controller for the YAH-64 Advanced Attack Helicopter (AAH) is provided by a Sperry Flight Systems designed and built SDP-175 microprocessor. The processor is located in the copilot-gunner (CPG) remote

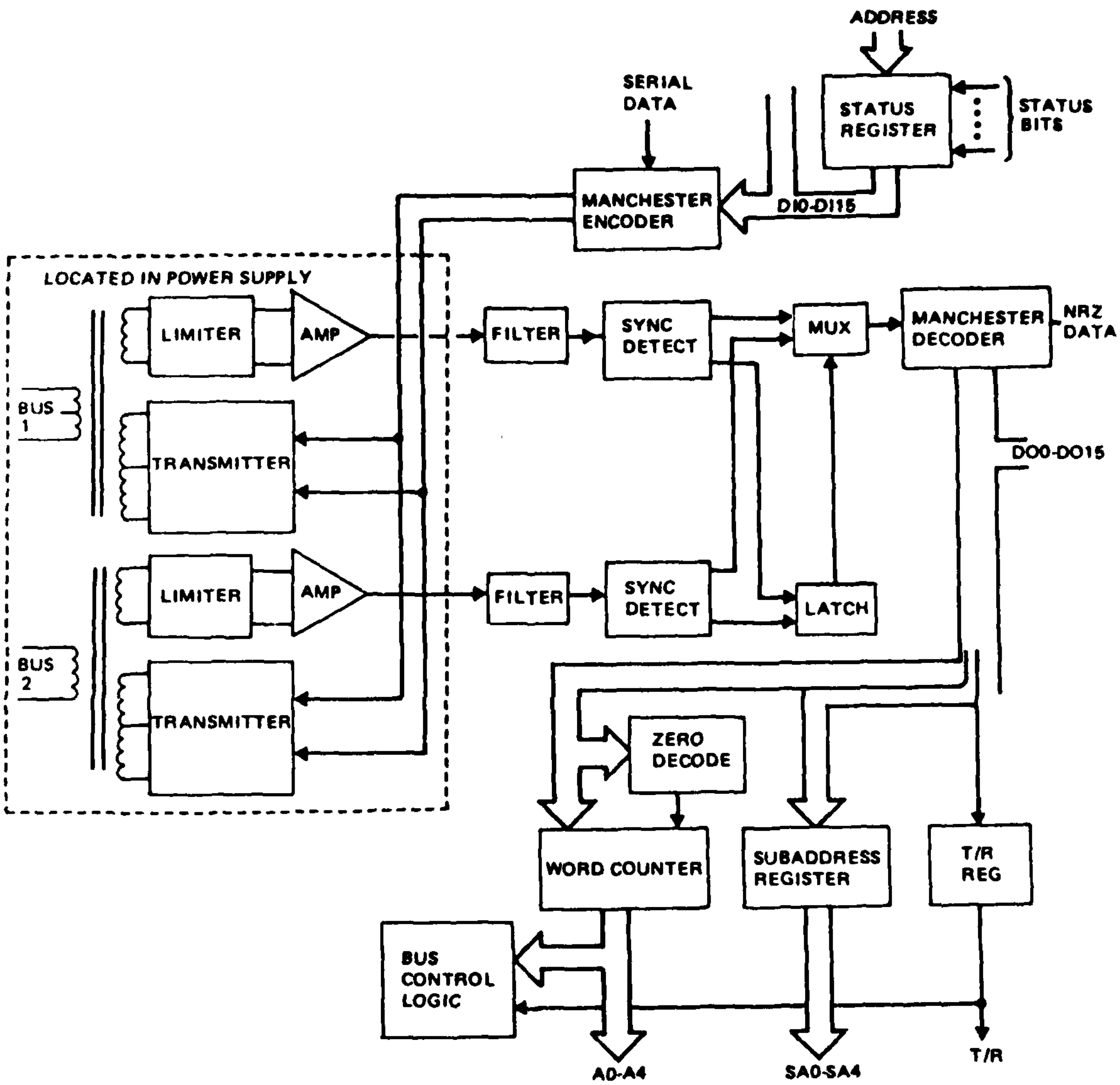


Figure 4.5-17. MRTU Bus Interface

terminal unit and is capable of performing a degraded mission function, as well as backup bus control, upon loss of the fire control computer (primary bus controller).

The backup bus controller is unique in that it is located in the same housing as an RT but is functionally separate from the RT. The functions are split between RT control and backup bus control in such a way that the RT does not know that the backup bus controller is residing in the same

unit. The functional separation of RT and bus control allows the other to operate in the event that one fails, barring failure of a physically shared component such as a power supply or the bus interface electronics.

As illustrated in figure 4.5-16, the backup bus controller consists of two basic parts; the SDP-175 microprocessor and a bus control unit. The SDP-175 is made up of four 2901A 4-bit/slice chips. It has 2K of RAM, 12K of PROM, and all of the necessary instruction decoding and multiplexing circuitry. The CPU is a 2's complement, fractional processor with 96 basic instructions and 6 special instructions to aid in bus control software. The processor provides for nine addressing modes, uses a pushup stack, and has eight fully vectored interrupts. Instructions include bit manipulation, Boolean functions, and arithmetic instructions that include double precision add and subtract, signed multiply, and unsigned divide.

The bus control processor allows the SDP-175 to do bus control operations with minimal CPU time. The bus control unit uses microinterrupts to advise the CPU when it needs to transmit a word or when it has received a word. The CPU microcode will then service the bus need. Therefore, the CPU can operate, with the bus control unit, at maximum CPU speed and thus is not affected by bus delays. This results in the CPU needing only 10% of its processing time to service the bus, which leaves 90% of its time for other software operations.

The bus control unit controls the bus by organizing commands to various RTs into one or more lists. Each command is followed by a word containing the number of words to be sent or received on the bus. Following the word count are the data to be sent and empty locations for the status response and data (if applicable) from the RT. A software command starts the interface and the first word is transmitted onto the bus. During this transmission time, the CPU is free to do other tasks. When the transmission is complete and the transmitter is ready for a new word, the bus control unit will generate a microinterrupt to the CPU, which will transfer the next word in the list to the bus controller and then return to its previous task. When a word is received from the bus, the bus control unit will save the word and generate a microinterrupt to the CPU, which will take the word, place it in an empty space in the list, and then resume its previous task. These CPU operations are controlled by microcode on an interrupt basis. They are executed between main program instructions and are, therefore, transparent to the programmer except for a 10% reduction in CPU execution speed during backup bus control.

4.5.3.4 Flexible Multiplex Terminal Interface

This section describes the architecture, features, and operation of a family of flexible multiplex terminal interfaces (FMTI) developed by SCI Systems Inc. This family is generally referred to as bus control units (BCU) but is actually a family of multifunction bus interfaces, each tailored for operation with a specific host processor. The BCU design provides a data bus interface unit with direct memory access, programmable interrupts, built-in-test, and programmable operating mode. BCU functions are shown in figure 4.5-18.

A BCU can operate as a 1553A bus controller, remote terminal, or bus monitor. Another available feature is simulation of all RTs on a given bus (up to 32). It is not implemented on all BCUs because of space constraints.

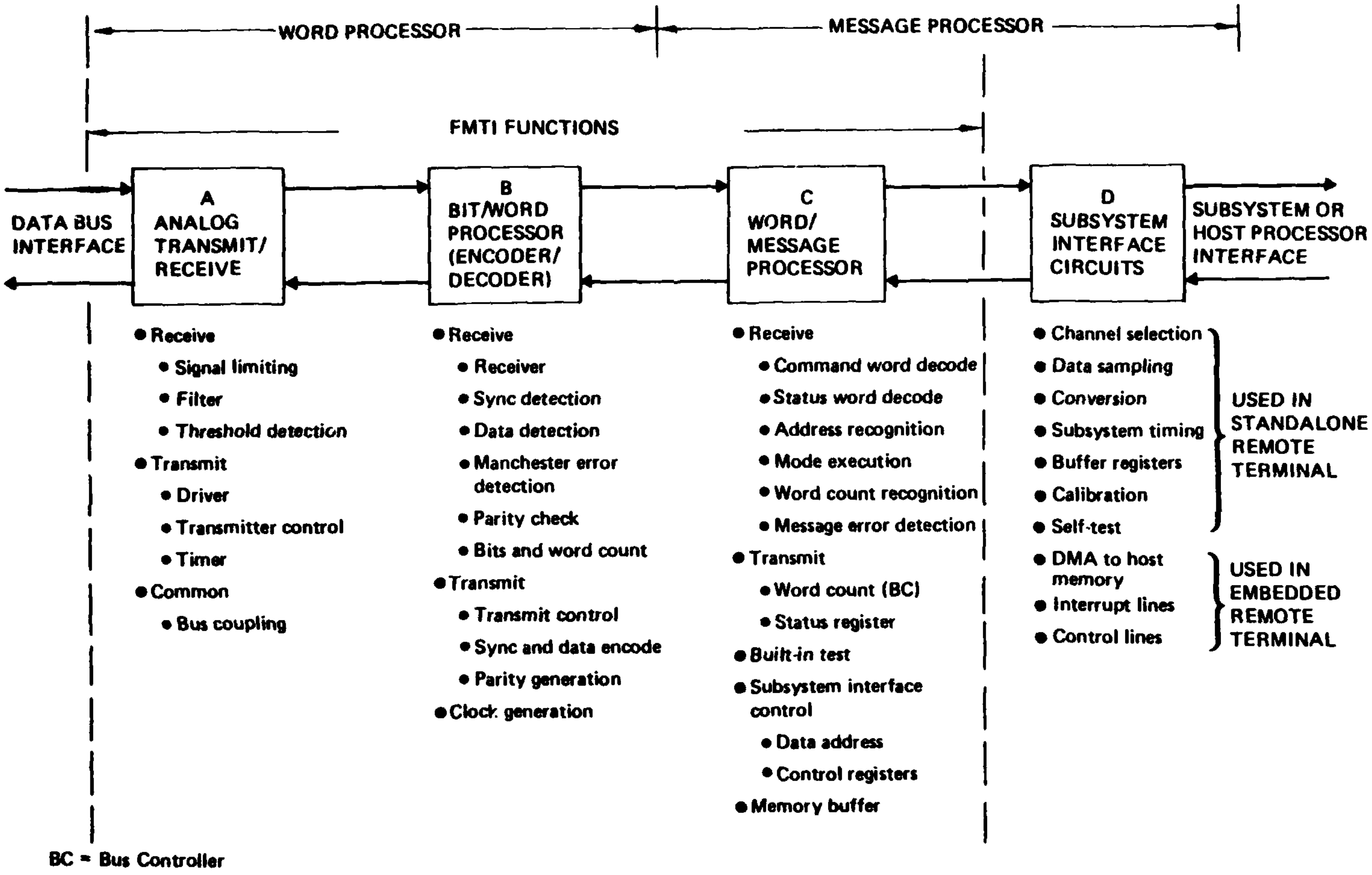


Figure 4.5-18. FMTI Functional Elements

The BCU design employs a bipolar bit-slice microprocessor for superior flexibility. The standard operating modes occupy approximately 800 words of microcode, while the architecture allows use of up to 4096 words. All operational features are firmware controllable. As a result, the upgrade to 1553B can be accomplished by a straightforward microcode update. Firmware commonality among BCU family members is greater than 90%, with common source code for all bus processing modules. This results in more stable microcode, since a "bug" fixed in any one application will be automatically applied to all BCUs.

4.5.3.4.1 Functional Description

A BCU interfaces a host processor to dual 1553A data buses. It has two basic functions: communicating with the host processor via programmed I/O (PIO) and interrupts and transferring data to or from the 1553 bus via DMA. Normally, these functions are mutually exclusive -- the BCU cannot be initialized or interrogated while engaged in bus activity. However, the BCU status register can be read and a shutdown request issued at any time by the host processor. Also, device flags can be tested, if provided in the host's architecture.

Before it can commence data transfers, the BCU must be initialized. This involves setting the contents of at least two internal registers (operation mode and activity pointer) by PIO commands. Other registers may be required in certain operating modes.

After initialization, the interface operates by interpreting the contents of a series of channel control blocks (CCB) obtained from the host memory via DMA. Each CCB, containing interface directives, 1553A test, and a link to the successor block in the chain, provides sufficient information to enable the BCU to operate without further host intervention. The BCU then executes the transfers specified by successive CCBs until the chain is exhausted or an error is detected in a bus transmission. In either case, a maskable interrupt is generated by the BCU, and bus data transfer ceases.

Both the host device code and the 1553 RT address can be specified by field-alterable straps. This is useful when multiple BCUs are installed in a single host. A default operating mode can be selected on the board, to facilitate the operation of multiple units on the same bus with common software. All switch settings are accessible to the driver software.

When operated in the 1553A command/response mode, the BCU can act either as a bus controller, in which it initiates all information transfers, or as one or more remote terminals, which can exchange data with the bus controller or another RT. If the multiple RT option is included, however, a given BCU cannot function as both the sending and receiving RT in a terminal-to-terminal transfer.

4.5.3.4.2 Architecture

The BCU hardware is organized into three primary elements: a serial digital interface with dual 1553A data buses, an internal 4 MHz 16-bit microprocessor and associated logic, and a parallel interface with the host processor. Figure 4.5-19 is a block diagram of the BCU showing the major functional blocks.

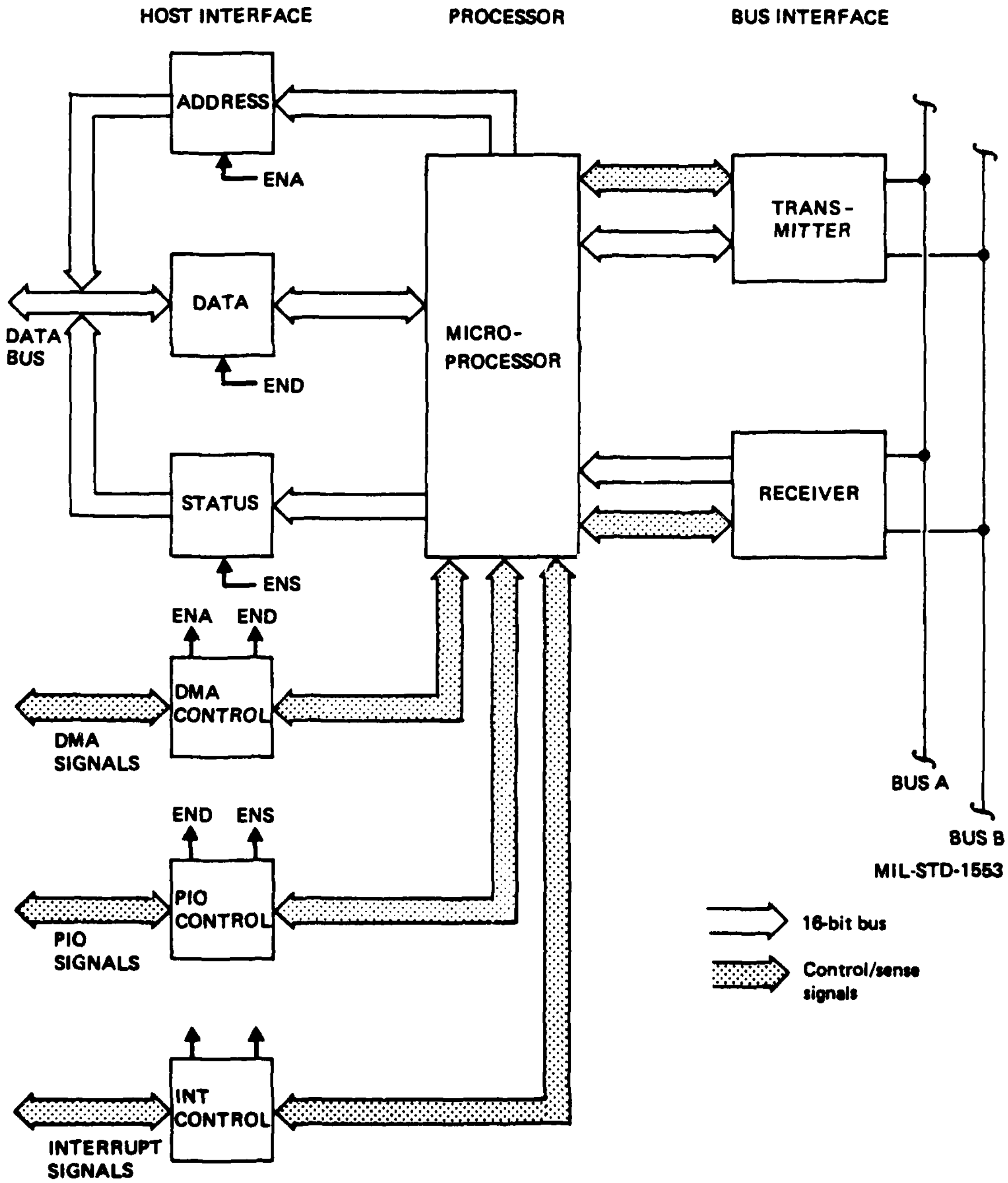


Figure 4.5-19. BCU Block Diagram

Host Interface

The host interface contains three full-width registers that are shared among DMA, PIO, and interrupt (INT) functions. This allows DMA operations to use the address and data registers and PIO reads to access the status register while the BCU is executing CCBs. Obviously, the DMA, PIO and INT control logic must be tailored to the particular application, but this involves less than 20% of the BCU hardware design.

DMA control is performed by independent hardware on all BCUs because the timing requirements exceed the response time of the microprocessor. Control information consists of DMA read and DMA write commands and a busy flag. BCU operation is largely unaffected by changes in DMA latency or transfer rate. Long latency times (19 us max) can be tolerated in the BC mode at the cost of longer intermessage gaps. Latency times for successful execution of the RT mode can be no greater than 11 us. If a DMA request is not honored in time, the firmware will request an interrupt after setting the appropriate bit in the status register.

PIO control is performed in firmware wherever feasible to reduce parts count. Obviously, the status input and shutdown commands are decoded in hardware to permit microprocessor independence. The PIO software interface is elegant on asynchronous machines, such as the PDP/LSI-11 family. In synchronous environments (NOVA, V77) the software interface is nonstandard, because BCU status must be tested before issuing a new PIO request to avoid possible overrun conditions. Many architectures include a series of device control and sense flags that can be directly used for this purpose.

Interrupt control is also achieved by dedicated hardware because of response time constraints.

Processor

The microprocessor employs bipolar bit slices (2901) to achieve 4 MHz instruction cycle times. It features a pipeline register at the PROM output, full carry lookahead, 16-bit shift/rotate operations, and a 32-word mask PROM to allow single bits or selected fields to be isolated in a single microcycle. Microcode size is 1K by 40 bits, with expansion possible to 4K words, because a 12-bit instruction address space is standard.

Two classes of microinstructions redefine the 40 bit instruction word. ALU instructions use 39 bits to specify operation code, register selects, data input source, output destination, mask selection, and shift type. They pass control only to the next sequential microinstruction. Sequencer instructions use between 28 and 32 bits to control next address generation. The full complement of 2910/2911 address control orders is supported, including two- and three-way conditional branches, subroutine nesting to four levels, conditional looping to a count of 1,023, and vector branches.

An assembler and support software package was constructed to aid in firmware development. The assembler features an easily comprehensible input format; the resulting code can be followed without consulting detailed flow charts. The firmware is organized into functional blocks: PIO handlers, diagnostic

mode, BC mode, RT mode, and support subroutines are all maintained as separate modules. These are assembled separately and the relocatable object code is linked to form a load module and PROM programming tape.

1553 Bus Interface

The serial interface provides all functions specified by 1553A, including signal characteristics, bus loading, and data validation. The transmitter, under processor control, provides a 1553A output signal with specified amplitude, rise and fall time, output noise, and transmission rate into a twisted-shielded pair data bus with up to 32 remote terminals. The design provides for short-stub or long-stub operation as a strap option and provides transformer isolation to the bus. Two versions of the bus interface are available. The standard design, which can be radiation-hardened, consists of two receivers and a single decoder constructed from TTL medium-scale integration (MSI). A later design incorporates two Harris Semiconductor 15530 complementary metal, oxide semiconductor (CMOS) encoder-decoder LSI circuits to achieve active/standby operation, where a valid command word on one bus will abort a transaction on the second bus. Two transmitters are provided on all designs. Waveform shaping can be provided as an optional feature for those applications that require a sine wave on the data bus.

The channel control word (CCW) format is given in table 4.5-5. The most significant bit (MSB) of the CCW is the CCB skip bit, which when set, causes the BCU to ignore the current CCB and continue to the successor CCB. Bits 8 to 14 of the CCW control interrupt generation. The remaining bits specify error generation options, transmitter and receiver selection, and two bits of option code uniquely defined for each operational mode.

The second word in a CCB is always the link address. This gives the address of the next CCB in the chain. If the skip bit is set in the CCW, the BCU immediately chains to the successor CCB without executing the current one. Also, if the stop bit is set, the BCU requests an interrupt without executing the remainder of the CCB.

Off Mode

Off mode operation is used for diagnostic purposes and does not result in information transfer on the data bus. While the BCU is in the off mode, all internal registers may be read and written, and switch settings may be read. In this mode, the I/O data transfer to all internal registers may be verified, and the BCU can be initialized for the other operating modes.

Bus Controller

In this operational mode, the BCU acts as a 1553A BC with the capability to transmit or receive data from RTs or to initiate RT-to-RT transfers and optionally capture their data.

The BC CCB format is given by figure 4.5-20. In addition to the control word and link, the CCB header contains a pointer to the message data area and the number of data words to be transmitted or received. A zero in the word count field indicates that no words with data sync are to be transmitted or received. The retransmit count specifies the number of

Table 4.5-5. Channel Control Word

Bit 15 (MSB)	Skip this CCB if set.	
Bits 8-14	Interrupt generation. Generate a CPU interrupt as defined below:	
	Bit 14 set:	Interrupt CPU immediately.
	Bit 13 set:	Interrupt CPU if the BCU status register indicates a transmission error.
	Bit 12 set:	Interrupt CPU if the RT message error flag is set in the status word.
	Bit 11 set:	Interrupt CPU if RT terminal flag is set in the status word.
	Bit 10 set:	Interrupt CPU if a bit is set in 3 MSB of RT status codes.
	Bit 9 set:	Interrupt CPU if a bit is set in 3 middle bits of RT status codes.
	Bit 8 set:	Interrupt CPU if a bit is set in 3 LSB of RT status codes.
Bits 4-7	Error block generation. Generate an invalid message as specified below:	
	Bit 7 set:	Bad sync in data, command, and status words.
	Bit 6 set:	Bad parity in data words.
	Bit 5 set:	Bad parity in command/status words.
	Bit 4 set:	Inhibit transmitter timeout (shutdown).
Bit 3	Transmitter selection	
	Bit 3 set:	Transmit on secondary bus.
	Bit 3 clear:	Transmit on primary bus.
Bit 2	Receiver selection	
	Bit 2 set:	Receive on both buses.
	Bit 2 clear:	Reveive on transmit bus only.
Bits 0-1	Option code	

retries to be attempted in the case of transmission error. If the message is successfully transferred before the count decrements to zero, an error interrupt will not be generated, even if enabled in the CCW.

The stored message consists of a header containing the actual command word to be transmitted, followed by a word reserved for the received status response. In the case of RT-to-RT transfers, two command words and status areas are necessary. Message data words may follow the header or may appear anywhere in the host memory. Note that the data pointer in the CCB header must be the address of the first data word to be transmitted or must point to a buffer large enough to contain the received data words.

The function to be performed by the BCU in the BC mode is specified by the option code in the CCW. Codes are specified for BC-to-RT and RT-to-BC information transfer, as well as RT-to-RT with and without the copy option.

Remote Terminal

In the RT operation mode, the BCU acts as a 1553A remote terminal with subaddress recognition capability. The BCU can either transmit data to a BC or receive data from the BC, depending on the state of the T/R bit in the command word it receives. An internal register is provided to specify the maximum time (in microseconds) that the data bus can be idle without generating an interrupt. This feature is enabled in the CCW option field.

The RT CCB format is given in figure 4.5-21. The CCB header contains the CCW, link, and status word skeleton. The skeleton is OR'ed with the parity

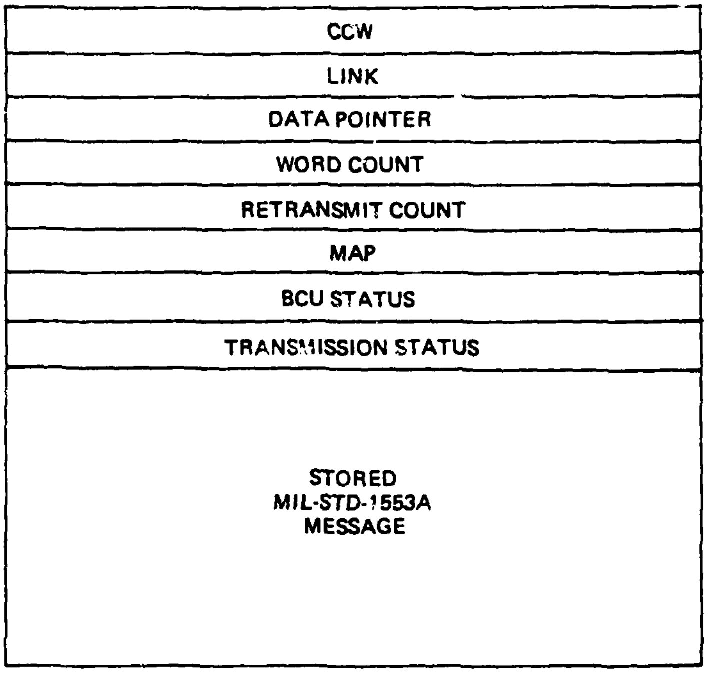


Figure 4.5-20. Bus Controller CCB Format

CCW
LINK
TEST 1 WORD COUNT
TEST 2 WORD COUNT
TEST 3 WORD COUNT
TEST 4 WORD COUNT
TEST 5 WORD COUNT
TEST 1 INPUT BUFFER
TEST 1 OUTPUT BUFFER
TEST 5 INPUT BUFFER
TEST 5 OUTPUT BUFFER

Figure 4.5-21. Remote Terminal CCB Format

error detect and RT bus address before being transmitted back to the BC. Thus, except for BIT purposes, the user should ensure that only the 10 LSB's are set in the skeleton word. Also, this skeleton is updated whenever any command word is detected on the bus; thus to change the transmitted RT status word, the user need only to store a new skeleton in the CCB.

The remainder of the CCB consists of 64 four-word entries, one for each subaddress T/R pair. The first word of the entry is the address of the data area and the second is a message counter. If the data pointer for a particular combination is zero, a BC reference to this combination will generate an interrupt. If the pointer is all ones, the reference will be ignored by the BCU, causing a message error to be detected by the BC. If enabled in the CCW, the message counter is decremented whenever a combination is referenced. When the counter becomes 0, an interrupt is generated. This is useful when a particular combination is to be acc ssed a specified number of times. The last two words of each entry are reserved for future expansion.

If the timeout counter is enabled, a deadline detected for longer than the specified period will generate an interrupt. The message counter and timeout updates are controlled by the CCW option code.

The BCU can operate in full compliance with 1553A requirements for RT operation, including the optional mode control feature. The mode control can be accomplished by zeroing the data pointer for subaddress 0 (SA0)to transmit and receive in the CCB. This will generate an interrupt whenever SA0 is referenced by the BC. Measured response time to a BC command word is 4.8 us.

It should be noted that the BCU will remain in a particular RT CCB for an indefinite time; it will enter a new CCB only when commanded by the CPU, unless the skip or stop bits are set in the CCW.

Multiple Remote Terminal

This mode allows the BCU to respond to command words for up to 31 remote terminals, as long as it is not participating in both the transmit and receive functions of a terminal-to-terminal message. It is an expanded form of RT mode operation and the same internal registers are provided. The CCB for the MRT mode consists of a CCW, a link word, and one or more RT CCBs. The format of the included CCBs is identical to that in the RT mode for standardization, but obviously the control words and links are ignored.

If a bus error is detected, the BCU status and transmission status will be written into the CCB, as well as the terminal address and subaddress of the RT detecting the error and the subaddress pointer. The CCB format is given in Figure 4.5-22.

Before initiating MRT operation, the number of terminals to be simulated must be set in an internal register. The BCU then determines the actual terminal addresses from the included CCBs.

CCW
LINK
RT COUNT
RESERVED
BCU STATUS
TRANSMISSION STATUS
RT IN ERROR
RT BLOCK POINTER
FIRST RT CCB
SECOND RT CCB
LAST RT CCB

Figure 4.5-22. Multiple Remote Terminal CCB Format

Diagnostic

This operation mode permits the CPU to verify the BCU internal data paths at three levels: DMA, transmitter-receiver logic, and transmitter-receiver analog signals. Five tests are performed in this CCB and the format is to read one or more words from the test input buffer, route them through the specified data path, and write them back to the test output buffer. The CPU can then compare corresponding pairs of input and output test words. In all tests, the output buffer size must be twice the input buffer size, because tests 2 to 5 also output the BCU transmission register, and test 1 adds the BCU status register. The size of each input buffer is specified in the CCB header and can range from 0 to 255. Note that this allows the transmitter shutdown feature to be checked by tests 4 and 5.

The CCB format is given in figure 4.5-23. Note that for test 1, the output buffer pair includes the DMA echo word and the BCU status register contents, and tests 2 to 5 write out the test result and the BCU transmission status register contents for every input word. A short description of each test follows.

a. Test 1: The microprocessor reads a word from input buffer 1 and writes it to output buffer 1. This tests DMA input and output functions, as well as internal data paths. If a noncompare results, the CPU can isolate the problem to either DMA input or DMA output registers via PIO tests if the microprocessor is functioning.

b. Test 2: Words from input buffer 2 are formatted by the transmitter control logic and routed to the receiver, bypassing the transmitter output. The words are formed with data sync. This test validates the transmitter and receiver logic for data type sync without appearing on the output data bus.

c. Test 3: This is basically the same as test 2 except that command sync is used. This test validates the transmitter and receiver logic for command type sync.

d. Test 4: Words from input buffer 4 are transmitted on the data bus with data type sync, with the receiver enabled. The received data is written to output buffer 4. This test validates the transmitter output stage and receiver front end for data type sync.

e. Test 5: This test resembles test 4, with the substitution of command type sync. This test validates the transmitter output stage for command type sync.

BCU Applications

The BCU design and architecture are compatible with most of today's 1553A avionic equipment. The microprogrammability features of the BCU enable the unit to be tailored to meet different user requirements primarily via firmware modification. Hardware changes are only required at the subsystem interface and, of course, repackaging is required to meet the user mechanical configuration. A photograph of a two-card assembly used in a ruggedized military computer is shown in figure 4.5-24.

CCW
LINK
TERMINAL ADDRESS/MAP
STATUS WORD SKELETON
BCU STATUS
TRANSMISSION STATUS
SA IN ERROR
SA BLOCK POINTER
(SA0 RECEIVE) MESSAGE POINTER MESSAGE COUNTER BCU STATUS TRANSMISSION STATUS
(SA31 RECEIVE)
(SA0 TRANSMIT)
(SA31 TRANSMIT)

Figure 4.5-23. Diagnostic Mode CCB Format

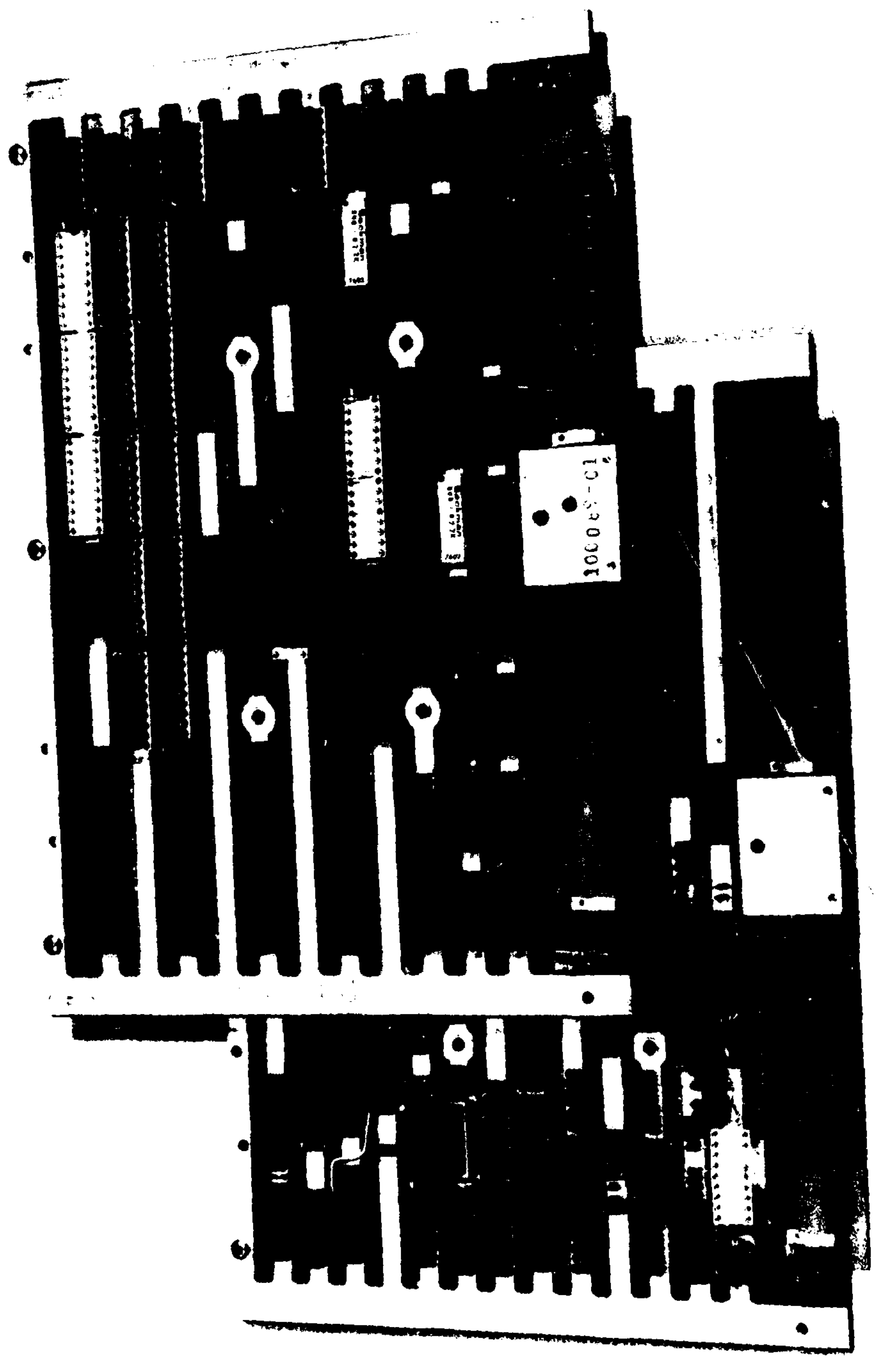


Figure 4.5-24. Two Card BCU Assembly

4.5.3.5 Remote Terminal Embedded in Subsystem

A recent development in remote terminal hardware involves modules and devices that, by simple pin programming, can be configured to different aircraft and weapon subsystems that employ various versions of MIL-STD-1553. These modules incorporate LSI chips and hybrid devices. The front end module (FEM) by Hughes incorporates the functions as shown in figure 4.5-25.

The front end module was designed to meet the following requirements:

a. Programmable to work with existing MIL-STD-1553 systems and new MIL-STD-1553B applications

b. Provides all its own intelligence and requires no support other than data from a host subsystem

c. Adaptable to both single and dual bus systems without modifications to the FEM

d. Full redundancy provided by the FEMs in a dual bus RT configuration so that no single failure will disable the RT

e. Produced using LSI and hybrid technology for optimum adaptation (size and power) to embedded RT installations

f. Provides a standard interface (between FEM and subsystems) that supports all functions of both AMUX and EMUX type subsystems

4.5.3.5.1 Front End Module Description

A front end module, shown in figure 4.5-26, consists of a bus interface hybrid, transformer, crystal oscillator, and line fault resistors. The bus interface hybrid contains a transmitter-receiver and two LSI chips; an encoder-decoder chip and a controller chip. The encoder-decoder chip is silicon on sapphire (SOS)-CMOS, operating with a 16 MHz clock to meet the McDonnell-Douglas Corporation (F-18) specification (A3818). The controller chip is standard metal gate CMOS, incorporating all the logic to operate as a standalone RT without need of the host subsystem.

A FEM has the versatility to work with any bus-subsystem configuration such as a single MIL-STD-1553 bus, a dual bus with common path to the subsystem and a dual bus with a subsystem that has a dual I/O interface. These configurations are shown in figure 4.5-27, (a), (b), and (c), respectively.

4.5.3.5.2 Interface Description

The FEM design can best be described by reference to figure 4.5-26, which shows the FEM accompanied by a description of the functional interfaces. Figure 4.5-28 depicts a representative dual bus RT showing the five interfaces involved in the operation of the RT.

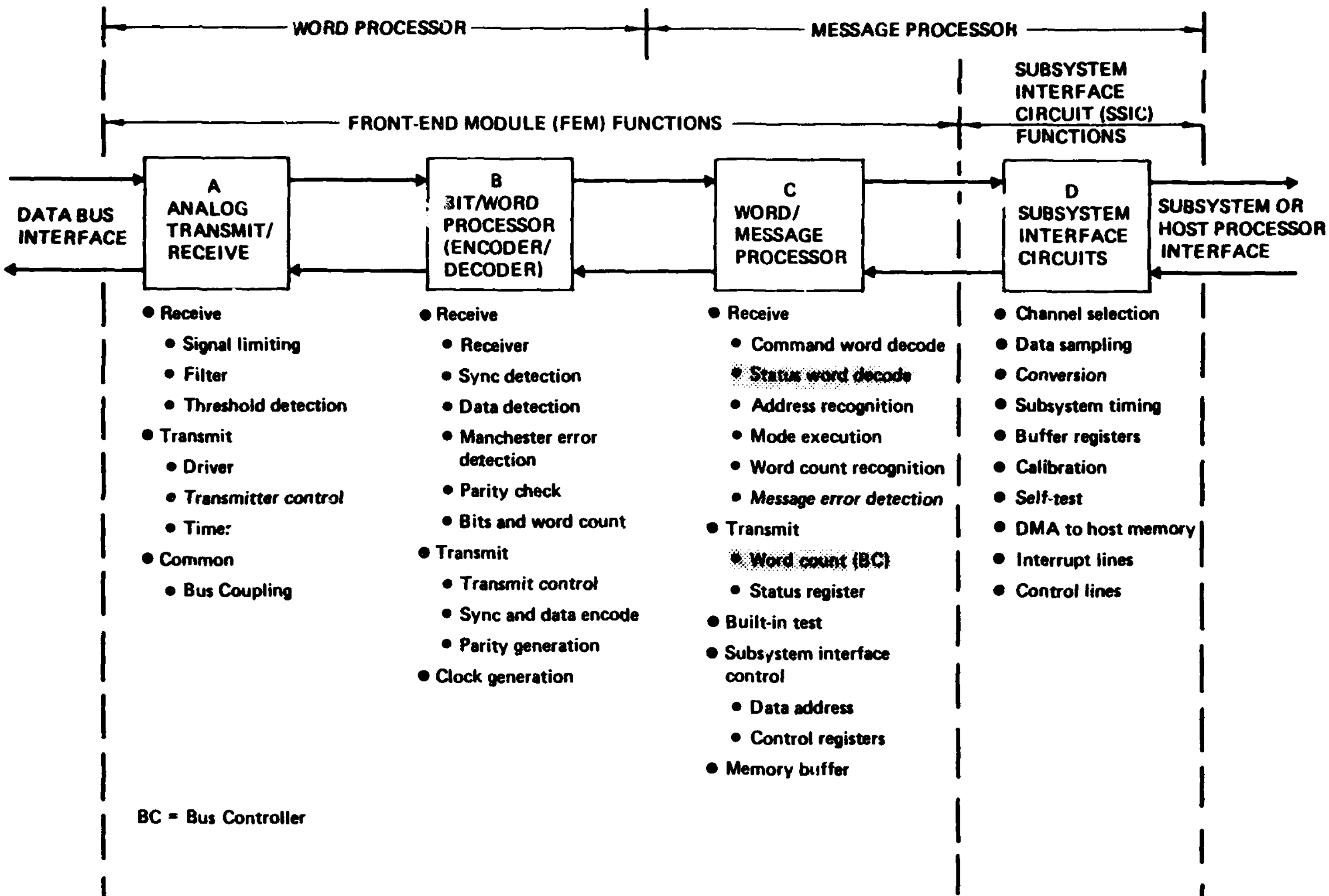


Figure 4.5-25. Remote Terminal Embedded in Subsystem—Functional Elements

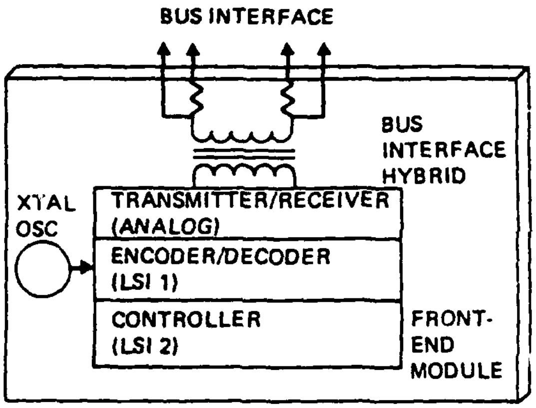


Figure 4.5-26. Front-End Module

Remote Terminal to Multiplex Data Bus Interface (1)

This interface is defined for each air vehicle and weapon system multiplex application based on which version of MIL-STD-1553 (i.e., MIL-STD-1553(USAF), MIL-STD-1553A, and MIL-STD-1553B) is employed and therefore is not discussed.

Programming Interface (2)

The FEM can be programmed for different MIL-STD-1553 formats and protocols required by various vehicles' multiplex systems. The programmable functions are made available on pins at the FEM or on the printed circuit board on which the FEMs are mounted. Programming is accomplished by connecting RT signal ground to the desired programming pin (logic 0) or leaving the pin nonconnected (logic 1). The interface provides programming capabilities for F-16 format and protocol, F-18 multiplex specification requirements, and MIL-STD-1553B applications. The programmable functions are as follows:

a. Protocol Select -- Two binary coded pin inputs are provided for selecting the protocol used for one of three MIL-STD-1553 applications; F-16, F-18, and MIL-STD-1553B. Differences in protocol and format occur

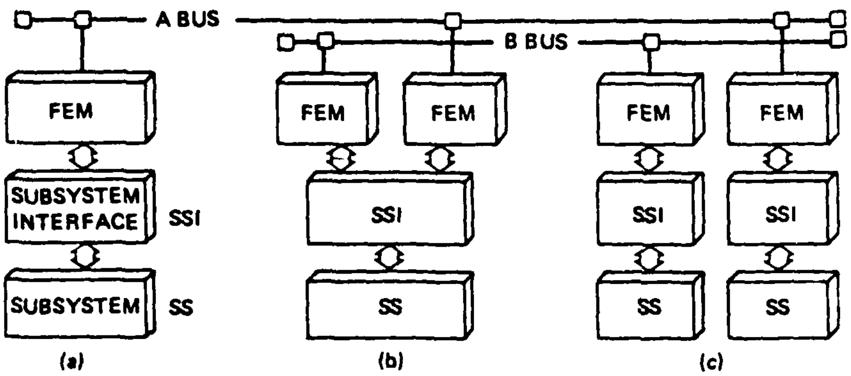


Figure 4.5-27. Variable Bus and Subsystem Configurations

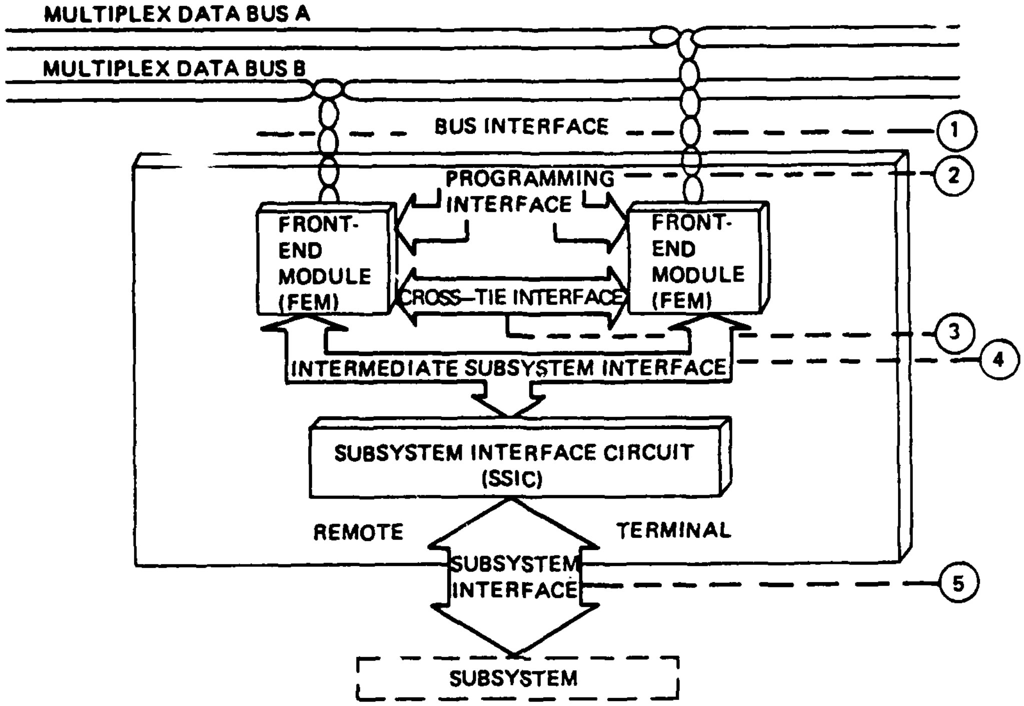


Figure 4.5-28. Representative Remote Terminal (Dual Bus)

in the area of status bit definition (see table 4.5-6), resetting of status bits, data handling under error conditions, and mode code assignments (see table 4.5-7).

b. Broadcast Disable -- Provided for MIL-STD-1553B applications. One input allows the broadcast functions of the terminal to be completely disabled.

c. Mode Code Designator -- One input is provided to define the subaddress mode field either as 00000 or 11111 for the mode code designator.

d. RT-to-RT Response Time Select -- One input selects the maximum time interval (either 7.5 or 12.5 us) the receiving RT will wait for a status response from the transmitting RT once an RT-to-RT command sequence has been detected. This function is incorporated to facilitate the difference in RT status response time between the various multiplex specifications.

e. Up-Down Word Count Select -- One input determines if the message word count field controller input to the subsystem should be incremented or decremented by one for each data word received or transmitted. DMA addressing of the host subsystem memory may use an incremented count, and other EMUX applications use a decremented count.

f. RT Address Select -- Five inputs are provided for the selection of the unique RT address.

Table 4.5-6. Status Bit Definition

Bit No.	F-16	F-18	1553B
9	Parity error	Message error	Message error
10	Instrumentation	ND	Instrumentation
11	Data quality	ND	Service request
12	NA	ND	ND
13	NA	ND	ND
14	NA	ND	ND
15	Broadcast command received	ND	Broadcast command received
16	Mode command received	ND	Busy
17	Bus B shutdown	ND	SS flag
18	Bus A shutdown	ND	Dynamic bus control acceptance
19	RT status	RT flag	RT flag

ND—not defined
NA—not applicable

Table 4.5-7. Mode Code Assignments

Command	F-16	F-18	1553B
Transmit status	X	ND	X
Transmitter disable	NR	ND	X
Transmitter enable	NR	ND	X
Reset RT	NR	ND	X
Synchronize	NR	ND	X
Reset timer	X	ND	NR
Inhibit T/F flag	NR	ND	X
Override T/F flag	NR	ND	X

ND—not defined
NR—not required

Cross-Time Interface (3)

A single-bus system application can be serviced by one FEM. In a dual-bus application, two FEMs are necessary. With two FEMs, operational redundancy and RT sectional control are made available at the subsystem interface. Carrying redundancy to this depth requires that the two FEMs have a means of cross-tie control to effect the inhibit or enabling of one another. Two critical intertie functions, "reset" and "valid comment received," are fail-safe; that is, no single point failure can knock out both FEMs, because the FEMs using the functions are looking for an edge (edge detection) rather than a single level (high or low). Other intertie signals are "transmit disable" and "bus shutdown" functions.

Standard Intermediate Subsystem Interface (4)

In the FEM design, this interface is very important because it affects standardization of the RT. At this interface, shown in figure 4.5-29, the necessary functions are present to enable any avionics-type subsystem to be serviced via bidirectional data transfer, or an electrical airframe-type RT to control the programming of its I/O signal conditioning circuitry. This design concept eliminates any redesign of the FEM. This interface, with the subsystem interface circuitry (SSIC), buffers the varying functions required by different host subsystems. In most dual-bus applications, the interface functions will be hardwired ORed from the two FEMs to the appropriate SSIC. Figure 4.5-29 shows that portion of the intermediate subsystem interface that provides bidirectional data exchange (via DMA) with an avionic communication system that incorporates a microprocessor. The interface functions labeled "provisional" are primarily used for implementing an RT to control the processing of EMUX-type signals (e.g., analog, discrete, and synchro). Of particular interest are the seven status lines made available at the interface, which can be used, depending on the multiple status word format required, to configure the status word bits for the particular MIL-STD-1553 type system being employed on the vehicle.

Remote Terminal-to-Subsystem Interface (5)

This interface will be different with each type of RT subsystem application because each function is determined by the I/O characteristics of each subsystem. A minimum of additional circuitry (SSIC) is necessary to interconnect the RT and the subsystem(s). In an avionic application, the interface would consist of a bidirectional data bus, addressing bus, and handshaking functions necessary to control and program signal conditioning modules. Figure 4.5-29 shows an RT-to-subsystem interface for a communication processor employing direct memory transfer to the subsystems.

4.5.3.5.3 Typical Remote Terminal Layout

Figure 4.5-30 shows a representative dual channel RT that is embedded in an air vehicle avionic LRU. Instead of using two FEMs, the devices normally contained in the FEMs (i.e., bus interface hybrid, transformer, oscillator, and resistors) are laid out on a printed circuit board (PCB). This configuration minimizes PCB area. The two bus interface hybrids provide the common, tristated, interface to the subsystem interface hybrid. This later hybrid then provides the signal path interface to the avionic subsystem. The RT as described makes greater use of the LSI and hybrid devices to achieve lower volume and power usage.

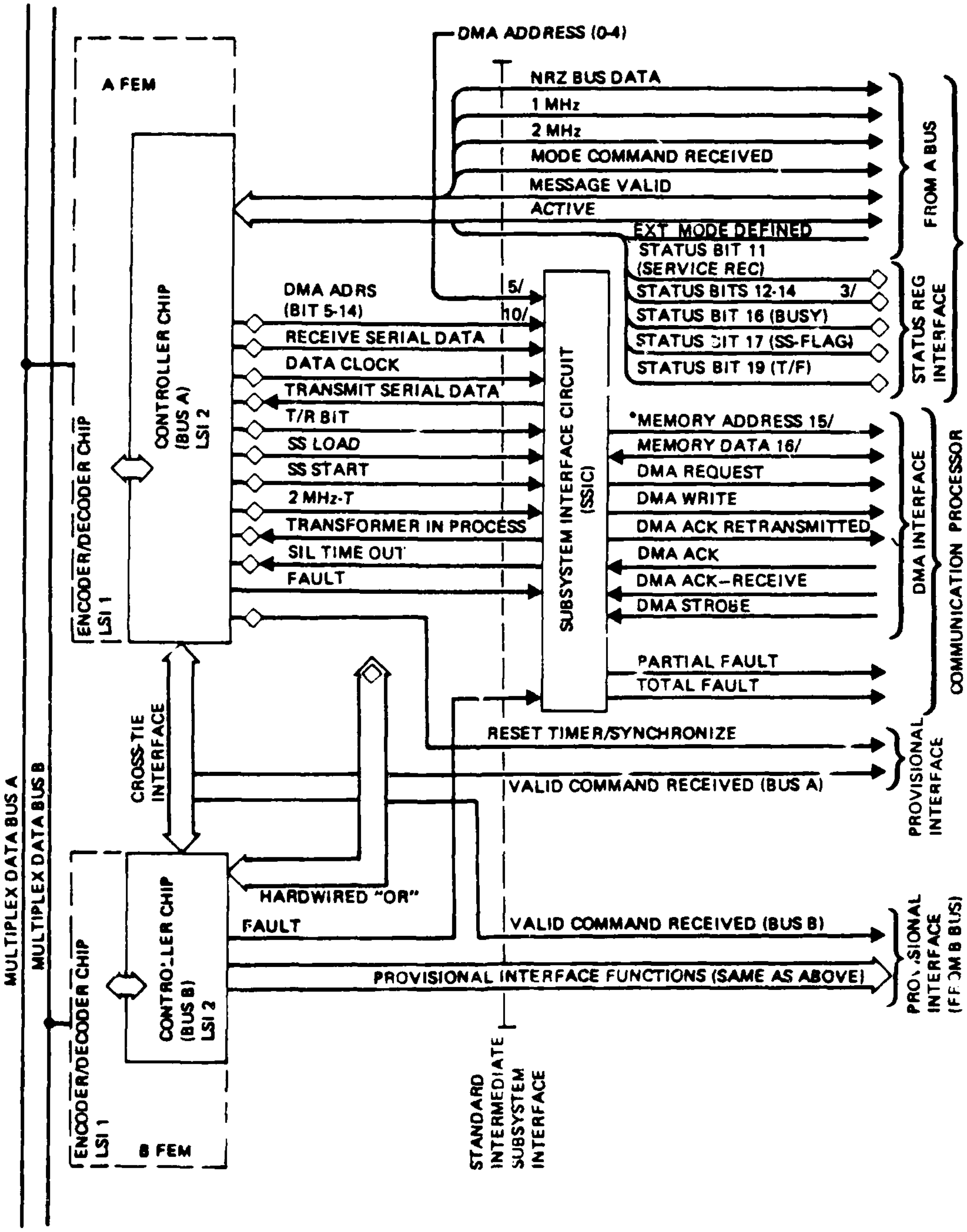


Figure 4.5-29. Dual Bus Terminal (Showing Intermediate and Direct Subsystem Interfaces)

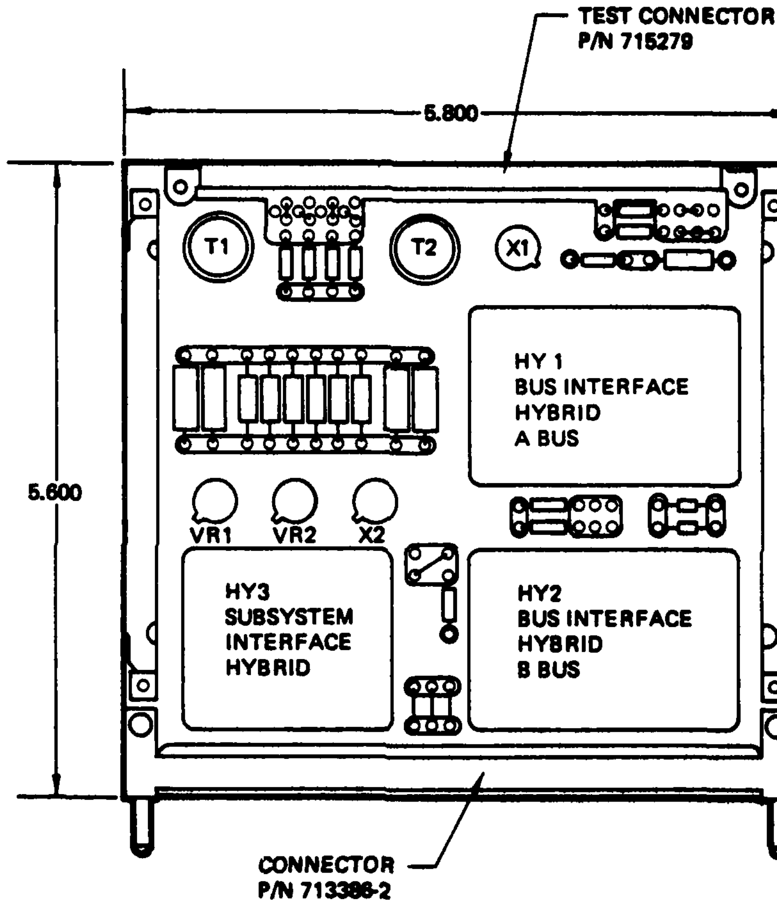


Figure 4.5-30. Dual-Channel Remote Terminal—Avionics Application

CHAPTER 5

SOFTWARE DESIGN

Table of Contents

Table of Contents (Continued)

List of Figures

1553B Figures

List of Tables

5.0 SOFTWARE DESIGN

This section deals with the techniques and considerations of software design for an avionic system using 1553 data buses. The discussion is concerned with the software that controls messages on the bus rather than application software. The term "control" software includes the avionic system executive, bus control, and error handling. The term "application" software includes such functions as navigation (dead reckoning, aided inertial dead reckoning, navigation sensor management) fire control, weapon delivery, and communication control. The interface of application functions with system control is included and is discussed from the point of view of segregating all supervisory functions from application software. "System" is defined to mean both the multiplex system and the avionic system, and the context will be made clear by the use of the terms "multiplex system" and "avionic system." Obviously, the multiplex system software must support the performance of all required avionic system functions.

System control considerations (e.g., what knowledge is needed) that affect the software design are discussed first, in section 5.1. Following this, section 5.2 describes the control of application software in a multiplex system. summarizing the functions of the control software. Section 5.3 discusses the multiplexing impact on application software. Section 5.4 discusses the support software that is unique to multiplex systems. Section 5.5 presents an example of bus control software in an attempt to expose many of the nuances in the design of this software for a modern 1553 multiplex terminal unit. In section 5.6, the BIU interface assumed in section 5.5 is described in greater detail to add understanding to the programming of such an I/O device.

5.1 SYSTEM CONTROL CONSIDERATION

The software designer of a multiplex system is typically faced with the task of determining the total software requirements concurrently with system and hardware design. The software engineer must obtain the requirements for software (of which the system control will be least obvious) from several sources. "System control" means both multiplex system and avionic system monitoring for correct operation, error handling, and failure handling. System control considerations are derived from requirements allocated to software to support the missions, transfer data, control the data bus, handle errors and failures, and, in general, manage the interface of the multiplex hardware and software. These categories of system control are discussed in sections 5.1.1 through 5.1.5.

5.1.1 System Functions and System Architecture

Software design begins with the statement of the requirements for functions allocated to software and a description of the system in which the software will operate. The overall statements of software requirements for a multiplex system will be derived from examination of functions and data requirements in the following areas:

a. Subsystems connected to the multiplex bus (or buses). What is connected to a bus? What are the data paths in the avionic system using the buses? What redundancy of data paths has been provided? What redundancy and/or isolation of function and equipment is required to be

managed by software? Answers to these questions provide the overall context of avionic system operation.

b. Missions and modes of each mission. It is necessary to know the complete repertoire of missions, how these missions are supported by functions of the avionic system, and what particular functions are to be performed during each phase of the mission. These groupings of functions by phase are called modes. (Note that these system or sensor modes are not really related to MIL-STD-1553 "mode codes".) For example, weapon delivery mode is usually distinguished from waypoint navigation mode, even though navigation sensors may be used in each. Apart from the data transfers that are unique or common to each mode, the unique setup (initialization) conditions must also be known. In short, a functional description of each mission mode that includes all functions of the avionic system must be known. Especially important to look for are time-critical actions that may require unique modes dedicated to time-critical tasks or may require suspension of other tasks.

c. Functions of each sensor. Descriptions are required for the inputs that each sensor requires (including sensor control information), what processing (computation) of sensor data is required for all avionic functions, and what data the sensor provides to the avionic system. Sensor redundancy concepts and the way that data from redundant sensors will be used or reconciled need to be described. Sensor modes versus avionic, weapon, and flight control system modes must be described, as well as the interrelationship of sensors (e.g., inertial navigation update using another position-fixing sensor).

d. Functions of control and display. Descriptions are needed of the overall interface of controls and displays to the avionic system and of the control and display functions that depend on multiplexed data.

e. Other avionic functions. The power and advantages of multiplexing often are applied not only to the integration of sensors, processors, controls, and displays but also to more simple devices (switch positions, actuator positions, etc.) and discretes (dc or ac analog, etc). The overall use of this type of data in the avionics system must be described.

The descriptions of functions, computations, and modes provided by b through e will establish the overall use of the 1553 data bus. It is quite likely that additional dedicated discretes will be used in an avionic system (for example, in stores management for enable or jettison), and therefore the use of these should also be established and described.

<u>Bus Control Approach</u>

The 1553 standard requires a bus controller to initiate all data bus use. The question of "who is in control" is very important. Since the bus controller operation is the single point of avionic and multiplex system control, it should meet the redundancy requirements of the entire system. Answers to the following questions relating to the approach for bus control are essential. What is the overall approach to controlling the data transfers of the avionics system? What is the concept of redundancy with

respect to the bus control? Is it a duplicate capability or an alternative control capability, or an alternative controller managing limited, degraded, or backup capability?

Obviously, avionic system control cannot be defined or the requirements made clear without an overall understanding and description of system functions, system architecture, and system control design. To avoid unnecessary work, the requirements for avionic system control must be derived "top-down" from the weapon system functions, the sensors, the selected avionic architectures, and the functions allocated to or expected of software. The purpose of the preceding discussion is to call attention again to the important step of defining the overall requirements for software from the system requirements. Sections 5.1.2 through 5.1.5 consider that the system functions are known and delve into the control considerations from four interrelated points of view: data transfer control, multiplex system control, multiplex system and avionic system error control, and bus interface hardware control.

5.1.2 Data Transfer Description

It is necessary to define all avionic system data transfers that are implemented using the multiplex bus. An examination of these will establish control requirements. Each individual data transfer must be defined in terms of the following attributes for each mode:

a. Source, function
b. Destination, function
c. Data definition in 16-bit words
d. Iteration rate if it is periodic data
e. Conditional events if it is aperiodic data
f. Allowable latency
g. Conditional events related to the data paths
h. Error response characteristics

Determining the source and destination of each data item is the first step in the definition of the input and output of the functioning avionic system. Included in this description is the definition of the data in terms of 16-bit words because that is the medium of transmission over the bus. A data item may be an input to different functions according to the particular phase of the mission and may be transmitted at different frequencies. Although it is true that earlier system design included these data transmission requirements in the overall system architecture and hardware requirements, it is necessary to either know exactly or make provision for each data transmission at the level of the definition of each bit in each data word transmitted.

Most multiplexed avionic systems operate on fixed schedules of data transfers. The requirements for the scheduling come from the examination of the largest and smallest minimum iterations and allowable latencies. The slowest iteration rate, which is the least common multiple of the faster iteration rates, is normally defined as the major cycle (see fig. 5.1-1). Over the course of a major cycle, all periodic transmissions occur at least once and all periodic computations occur at least once. Some exceptions do exist if the iteration frequency is very low (such as Kalman filtering once

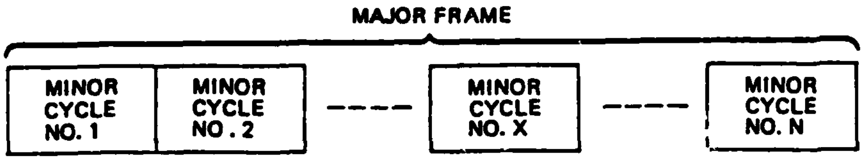


Figure 5.1-1. Major and Minor Cycles

per 6 sec, or periodic built-in-test functions once every 10 sec). The minor cycle is normally the frequency of the most rapidly transmitted periodic data. Typical major frames are 1 sec in length, while minor frame lengths can be binary (2^N/sec) or decimal (10^N/sec) with common values being 1/128, 1/64, 1/50 sec, etc.

For example, if the major frame is 1 sec long and there are 64 (2^6) minor cycles, then each minor cycle is 1/64 sec or 15.625 ms long. Each periodic message would occur at least once each major frame, up to a maximum of 64 times. If a transaction needed to occur eight times per second, it must occur during one of the first eight minor cycles (64/8 = 8) and every eight minor cycles thereafter. The minor cycle in which the first message occurs is known as the "phase," while the repetition rate is its "period."

In the example of a transaction occurring eight times per second, shown in table 5.1-1, if the first transaction occurred in minor cycle 3 (the phase), later transaction would occur in minor cycles 11 (i.e., 8 + 3), 19, 27, 35, 43, 55, and 63.

Aperiodic messages, while rare, are based upon conditional events and are used to initiate other conditional events. The conditional events relating to aperiodic messages might include a requirement to present a display to a crewmember within X-milliseconds of keyed in commands or keyed in requests for data. Another example of a conditional event relative to an aperiodic message might be a requirement to acquire data in a data buffer before it is lost because of the next input to the buffer. This case is typical for

Table 5.1-1. Message Frequency Table

Times per major frame transaction occurs	Period	Possible phases that could occur
1	64	1, 2, 3, . . . 64
2	32	2, 4, 6, . . . 32 . . . 64
4	16	4, 8, 12, 16, . . . 32 . . . 60, 64
8	8	8, 16, 24, 32, . . . 56, 64
16	4	16, 32, 48, 64
32	2	32, 64
64	1	1

keystrokes from keyboards. There may be a requirement for data acquisition within X-milliseconds (which may be a hardware-imposed characteristic).

The software designer needs to know the system data transfers so that both control software and application software can support the required transfers. Close association and cooperation of engineering groups responsible for defining the system functions, system architecture, and data transfers will reduce rework and errors. The software designer should participate in this phase of the effort. Once the data communications have been defined and the functions have been allocated (approximately) to processors, then the data can be grouped into messages. The data of the same frequency can be grouped into the same set of messages. Data that must be transmitted in the same period can be grouped. Several points must be made about the grouping of data into messages:

a. Do not attempt to group functionally dissimilar information together to minimize the overhead unless necessary.

b. Provide spare capacity in the message sizing and allocations to terminals (maximum message is 32 words and 30 subaddress). Just as functions grow during design and development, so do the communications between functions.

c. Bit packing of data greater than 1 bit should not be done unless necessary. Packing and unpacking takes both time and hardware complexity (e.g., 8-bit analog data should not be packed 2 to a word or discretes packed in with analog data).

d. Attempt to isolate data (functions) that are likely to change over the life of the avionic system from other basic avionic messages to allow for the minimization of disruption of messages because of future modifications.

5.1.3 Multiplex System Control and Avionic System Control

MIL-STD-1553 requires that the multiplexed data transfers be initiated by a bus controller and that each data transfer on the bus be commanded by a bus controller, followed by a response, in normal operation. The only exception is that a bus controller may command a broadcast, which does not require a response. In all cases, the actual transmission is either a combination of the 1553 protocol and avionic data formatted into 16-bit words or it is a transmission of 1553 protocol information without avionic data.

The response time allocated to the remote terminal's normal response to a bus controller's command and the data rate and format of 1553 bus have imposed requirements such that the bus interface unit is distinct hardware and a distinct functional part of remote terminals and bus controllers. The software designer must know the specific characteristics of the bus interface hardware design to understand the roles of the processor and the bus interface hardware. Section 5.1.5 discusses the effects of types of terminal hardware on software design. During bus operation, transmission failures may occur, subsystems that are connected to the data bus by a bus controller or remote terminal may fail, or the multiplex hardware itself may fail. Each type of failure will interfere with normal data transfers.

Section 5.1.4 discusses requirements for software design that must be defined for errors and hardware failures.

The balance of this section is limited to a discussion of normal control of avionics and the multiplex system. Normal control is divided into the following subtopics:

a. Types of data transfers and their impact on software design

b. Multiplex system control, using mode codes, and its impact on software design

c. Required capabilities of the system executive

5.1.3.1 Avionic Data Transfers and Avionics Control

The types of avionic data transfers that are defined and allowed by 1553B are briefly described below. Each data transfer will also include some multiplex system control information, but that part of the discussion has been deferred to section 5.1.3.2. Data transfers were discussed in section 5.1.2. The data will be a combination of avionics control-related actions, such as flags that indicate that a subsystem is initialized, or engineering measurements, such as pitch angle in radians. The multiplex bus is entirely appropriate to effect avionic system control. It also follows that the normal functioning of the avionic system and the monitoring of its status can and should be accomplished via data bus transfers. The types of data transfers and the implications for the software designer are as follows:

a. Remote terminal to bus controller. This type of data transfer is used to provide data to the bus controller. The bus controller is almost always a mission computer, fire control computer, navigation computer, etc. As such it requires data from several sensors like air data, inertial navigation system, inertial measurement unit, radio navigation, etc., to perform its assigned sensor computational functions. Therefore, it usually will also be assigned the function of bus controller and will initiate the requests to the remote terminals for the data that the processing software needs. The data needs of the mission computer's assigned processing tasks establish the requirements for data transfers to it from other sources on the bus.

b. Bus controller to remote terminal. Typically these types of data transfers are related to the role of the processor that has the bus control function. A mission computer may have the requirement to be the data source to devices, providing such data as position update to an INS, or the requirement to transmit display parameters to a graphics generator. The mission computer often serves as the processor to effect weapon system control, such as fire control, in which case it is controlling both the multiplex system and computing parameters for target designation, weapon initialization, etc. In this case, controller to remote terminal transfers are data transfers from the fire control computer to the remote terminals that contain the interfaces to target designators, stores, etc. These types of data transfer may also be used as a method of central distribution in which data are taken from a remote terminal, reformatted and retransmitted to other locations.

c. Remote terminal to remote terminal. The bus controller does not need to receive and retransmit all data even though it is in control of the bus. An important class of data transfers is the direct transfer of data from one remote terminal to another, which can be used if the processor that contains the bus controller is not involved in the processing of the data and if reformatting is not required. In avionic systems that employ more distributed processing (e.g., CADC, INS on the bus) the additional processing capability at those remote terminals can be used to select and format data for direct remote terminal to remote terminal data transfers.

d. Broadcast. The broadcast data transfer is an option of 1553B but not currently in general use in military airplane avionics. Broadcast allows the simultaneous transmission of the same data to more than one remote terminal. The broadcast information transfer format may be used for avionic data transfers when significant reduction in processing or bus message traffic is needed and the command/response validation feature of each message is not required. For example, broadcast of roll and pitch data for aircraft flight-path control to a dual-, triple-, or quad-redundant flight control system may serve to simplify both avionics and flight control software. The use of broadcast can also be effective when identical data must be transmitted to multiple devices, the latency of serial transmissions will not meet the computational requirements, and the command/response message validation feature (per message) is not required. Note that broadcast does not obviate the need for determination of status (e.g., message received) but that status cannot be determined by broadcast. Status can be determined by separate requests of the controller to remote terminals.

Each unique data transfer must ultimately be identified to the multiplex system hardware and software by its unique combination of terminal address and subaddress. It is this feature that establishes the requirement that the data transfers be organized into messages. Because the decoding of a subaddress (as well as the terminal address) is usually done completely in hardware, the assignment of subaddress for each unique data transfer or data block is normally a system task. This statement should not be interpreted as meaning the assignment of subaddresses cannot or should not be under software control. For example, software could be used to load a register with a set of subaddresses at the beginning of a particular minor cycle, even though the response time requirements of 1553B will not allow software decoding of each subaddress. It is common in the system design to prepare tables that define, for each subsystem on the bus, the complete data transfer specification into messages. Entries in the table are usually as follows:

a. Subsystem name, for example, INU, CADC, radar, radar display

b. Subsystem terminal address, viz, a 5-bit binary number, per 1553B, paragraph 4.3.3.5.1.2

c. Data block ID, viz, a reference to a detailed word-by-word description of the data

d. Subaddress, viz, a 5-bit binary number per 1553B, paragraph 4.3.3.5.1.4

e. Word count, viz, a 5-bit binary number per 1553B, paragraph 4.3.3.5.1.5

f. Refresh rate, for example, the rate at which the subsystem updates a variable

g. Transmit rate, viz, the intended rate, usually stated as a minimum value, at which the subsystem will be requested to transmit the data

Separate tables are required for transmit and for receive for each terminal, whether it is a remote terminal or a bus controller. If a terminal may also be a bus controller (such as a backup or an alternate), additional separate tables are required. (Such tables are an expansion of the data transfer description documentation, sec. 5.1.2). The impact on the software designer is to define all data blocks that will be transmitted or received under all system conditions, normal or abnormal. The control software will handle the data according to the data block definition. Therefore, an exact correspondence between the input and output of data blocks and the use of the data must exist.

Table 5.1-2 shows an example of some assigned data blocks to subsystem subaddresses and the definition of a single word from one of the blocks. Note from the example that each data block has a unique subaddress even if the word counts, refresh rates, and transmit rates are identical.

5.1.3.2 Multiplex System Control and Software Design

Every data transfer via the 1553B bus contains multiplex system control information in accordance with the 1553 protocol. Two options exist for the system designer to increase the control capability, which will affect software designs: (1) whether additional data transfers shall be used that are dedicated to the multiplex system management and (2) how much of this optional multiplex management shall be used. In addition to the status word, which is routinely received, 1553B provides for "mode control," which according to paragraph 4.3.3.5.1.7 "... shall only be used to communicate with the multiplex bus related hardware, to assist in the management of information flow, and not to extract data from or feed data to a functional subsystem."

The "types of data transfers" discussed in this section are "information transfer formats" according to figures 6 and 7 of 1553B, and are described in paragraph 4.3.3.6 of the standard. The definition of the words in the data transfers is in paragraph 4.3.3.5 of the standard. The most commonly used of these information transfer formats (really message formats) are the command/response formats of figure 6. Note that as a result of using any of these formats, the controller will receive status. Evaluation of the mandatory status word, if received, will establish whether operation of the multiplex system is normal and will indicate that no subsystem failures have been detected.

The evaluation of the status word, if received within the response time of 1553B (fig. 6), is divided between the multiplex hardware and software. Nine status bits are available for use, of which two are required and the other seven are optional. Multiplex hardware will evaluate the status word and, if none of the flags in the status word are set to logic one, normal operation is ensured. Optional status bits, if not used, are set to logic

Table 5.1-2. Example of Assigned Data Blocks to Subsystem Subaddresses

SUBSYSTEM SUBADDRESS/WORD COUNT/RATES

Transmit subsystem	Block ID	Subaddress	Word count	Refresh rate (time/sec)	Transmit rate (times/sec)
INU	I01	10000	28	50	50
	I02	10001	10	5.0	6.25
	I03	10010	8	5.0	6.25
	I04	10011	13	50	50
	I05	10100	13	50	50
	I06	11101	31	50	50
	I07	11110	32	2.5	6.25
CDU	P02	10000	4	5.0	6.25
	D01	00001	7	5.0	6.25
CADC	C01	00001	10	25	25
	C02	00010	9	25	25
	C03	00011	2	25	25

WORD 9—PLATFORM AZIMUTH DATA FORMAT BLOCK ID: I06-09

Data bit	Description
1	Sign bit
2	MSB (0.5π radians)
3	•
4	•
5	•
6	•
7	•
8	•
9	•
10	•
11	•
12	•
13	•
14	•
15	•
16	LSB

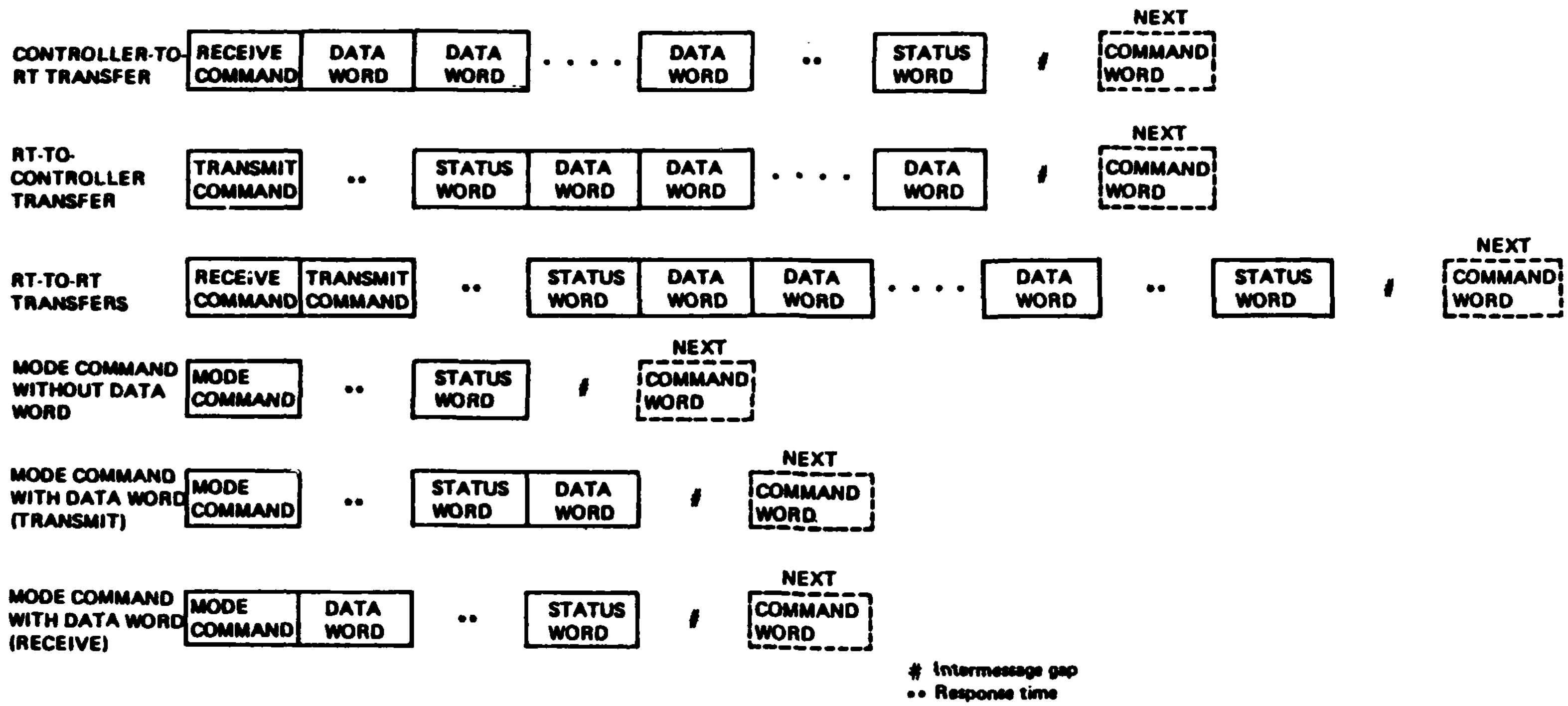


Figure 6 of 1553B. Information Transfer Formats

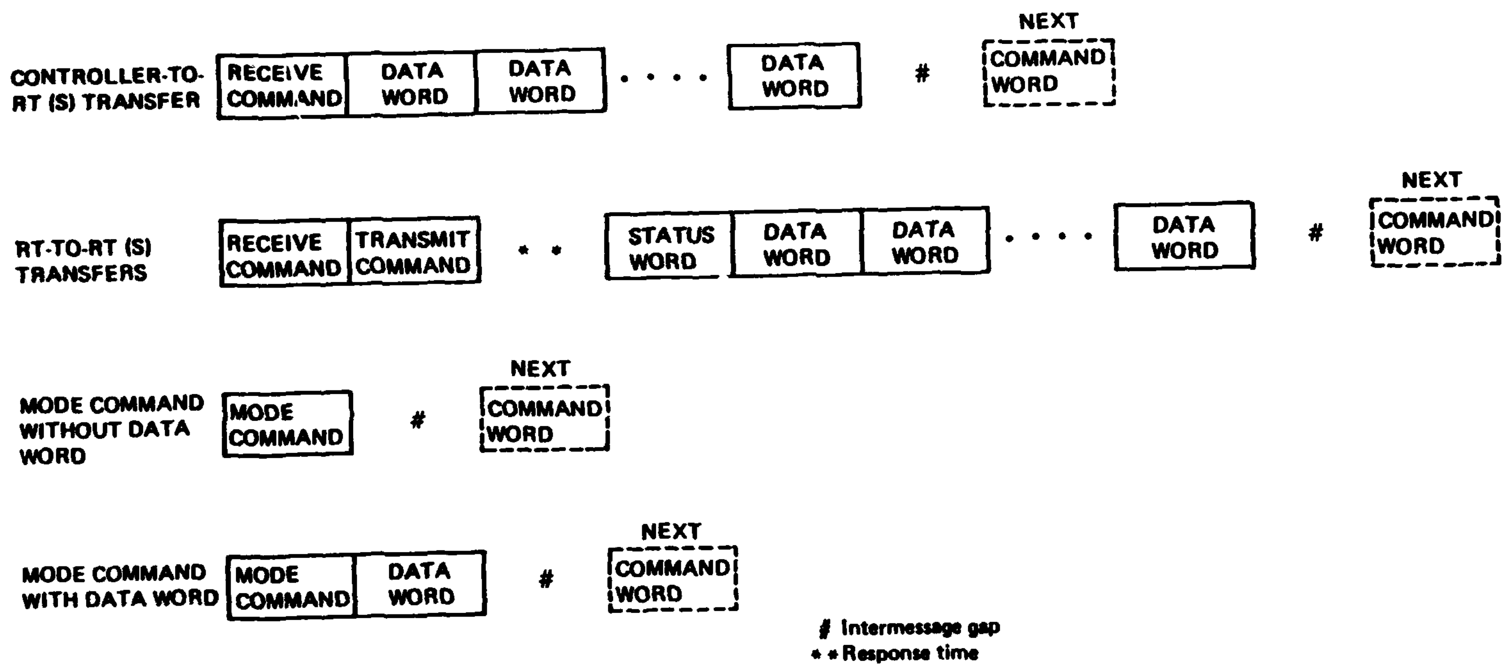


Figure 7 of 1553B. Broadcast Information Transfer Formats

zero. While it is conceivable that a remote terminal could evaluate its own abnormal status, 1553B does not allow the status word to be reset independently of bus controller action.

Once the status word is decoded, it will normally be saved by the bus controller (until the next status word arrives) for possible inspection by the software. If any of the bits is logic one, the usual hardware implementation is to cause an interrupt so that error detection software can be used to evaluate which status flags have been set. If the status word is not received, hardware will normally indicate to the software via an interrupt.

The majority of the optional mode codes assist in multiplex and avionic system management if flags in the status word are set, or if the status word was not received. The usefulness of the mode codes is quite dependent on the system architecture and the system control that is implemented. The 15 mode code definitions in 1553B, paragraphs 4.3.3.5.1.7.1 through 4.3.3.5.1.7.17 should be reviewed to determine if any are appropriate for a particular multiplex system application.

Should the optional mode codes that relate only to multiplex management be used? Again, this is a system question, but the software engineer is certainly impacted by the decision. If the optional mode codes are used, the software design will be more complex because interrupts will accompany their use. The use of the optional mode codes will provide significant capability to control and respond to unusual multiplex conditions (such as errors).

Consider the synchronize function enabled by two mode codes (see 1553B, pars. 4.3.3.5.1.7.2 and 4.3.3.5.1.7.12) differing only in that one is transmitted without a data word. Avionic data transfers are mostly periodic, and processors in an avionic system cannot operate synchronously without coordination. Also, the timing of data transfers to maintain the system periodicity requires coordination in systems that use two or more processors to distribute the computational load. In single processor systems with bus controller in the processor, system data transfer periodicities can be maintained using the timing of the processor to control data transfer event timing, because all data transfers are initiated by the controller. This example shows that "synchronize" use depends on system architecture. If "synchronize" is used, the software designer will have to access at least one c ɔck, in one processor, decode its output, and interrupt to transmit the command. The broadcast option may be used to transmit the synchronizing messages to all the remote terminals simultaneously.

5.1.3.3 Required Capabilities of the System Executive

Multiplex system control and avionic system control are accomplished via the executive. For an avionic system that uses 1553 multiplexing, there are very few of the executive functions that are inseparable from the primary task of avionics control and multiplex system control.

The following list presents required executive capabilities:

a. Startup. Define the actions required following power-on to the processors in the system. The state of the registers, the power-on interrupt location, and functions that need to be performed during normal startup must be specified.

b. Loading the program. Define the requirements for program load. Some systems use the multiplex bus for this function, by communication with an onboard mass storage unit. Some systems use a direct I/O interface with the mass storage unit. Still others have programs in ROM, obviating the need for program load.

c. Input and output management. Describe in detail the method for transferring I/O data between the processor and external interfaces using non-1553 and 1553 capability, initiating I/O operation and resumption of normal processing until I/O completion. The control of the data bus to detect and handle transmission errors must be also included.

d. Task scheduling and control. Define and describe in detail the methods used for (1) controlling the data interfaces between the executive and applications programs, (2) sequencing the periodic application programs, and (3) sequencing the aperiodic programs in accordance with their conditional events. The sequencing required by (2) and (3) above is usually handled by message lists for periodic applications with appropriate interrupts for aperiodic applications. A discussion of hardware implementations is prerequisite to an expansion of methods for sequencing applications (hence bus traffic), which will be deferred to section 5.1.5, "Bus Interface Hardware Control."

e. Interrupt handling and processing. Define all the interrupts, their priority, and what executive and applications programs they invoke.

f. Unscheduled event recovery. The typical requirement in this category is primary power interruption.

g. Reconfiguration. Requirements in this category fall into the following areas: (1) distributed processing systems where one or more processors fail, (2) backup systems when the primary bus controller processor fails, (3) failure of redundant subsystems, and (4) failure of nonredundant subsystems.

h. Special test functions

i. Shutdown

5.1.4 Errors and Hardware Failures

Provision must be made during system design to handle errors in data transmission, power transients, hardware failures, and data errors. Very few applications can be conceived of that might provide software error detection. A software error is any failure of the software to accomplish what its designers or coders intended it to accomplish. These error detection approaches are not well defined and therefore will not be included in this discussion.

The 1553 data bus, with its prescribed protocol and hardware characteristics, provides superior transmission error detection capability, but the standard allows the system and software designers to select the remedial course of action to be taken if an error is detected. The 1553 requirements that are applicable to this discussion are:

4.4.1.1	Terminal word validation
4.4.1.2	Terminal transmission continuity
4.4.3.1, 4.4.3.3, and 4.4.3.4	Remote terminal operation--acceptance and rejection of commands
4.4.3.5	Remote terminal operation--response of the status word after valid data reception
4.4.3.6	Remote terminal operation--suppression of the status word after invalid data reception
4.5.2.1.2.4 and 4.5.2.2.2.4	Noise rejection--maximum allowable word error rate

Determining the requirement for a response to a detected error is difficult, particularly for the system designer or system software designer, because there is no guidance in 1553 and no well-accepted guidelines for doing an analysis that shows that one set of responses is superior to another if mission completion success probability is the measure. That is to say, if mission completion success probability is computed for several candidate responses to detected errors, only those system actions that increase the probability should be considered for implementation. On the other hand, laboratory investigation (as in a hot bench) may be highly desirable to determine both the effect of a response and the cost in software sizing and timing to get the response. In order to better understand this problem, several typical errors have been postulated with examples of typical corrective action described (see table 5.1-3 and 5.1-4).

Errors and hardware failures can be classified into reasonably exclusive indications. These classifications and general requirements for responses that will provide guidance on the implications of responses to detected failures are discussed in the following sections.

5.1.4.1 Detected Message Completion Failures

MIL-STD-1553 defines word and message validation criteria, which were referenced previously. If the multiplex terminal hardware detects either an invalid word (1553B, par. 4.4.1.1) or a transmission discontinuity (1553B, par. 4.4.1.2), then the word and message are to be considered invalid. The standard does not specify what hardware characteristic or software process will determine that the already received word, words, or message is invalid and that it not be used. Nor is there a mandatory requirement in 1553 that any investigation at all be instituted on detection of an invalid word or message.

Table 5.1-3. Error-Determination Approach

Error identification	Failure classes	
	Bus system	Sensor
a) Message error	Transmission from bus controller to terminal was decode with error condition by receiving remote terminal	–
b) Busy	Remote terminal unable to transmit or receive data at this time	Remote terminal and sensor unable to transmit or receive data at this time
c) Subsystem flag	–	Sensor failure preventing proper sensor actions
d) Terminal flag	Remote terminal failure preventing complete action by terminal	Remote terminal portion of sensor interface has failure preventing complete action by terminal
e) Parity error (incorrect odd parity)	Error in status word; data not usable	Error in status word; data not usable
f) Improper sync	Unknown problem—ignore; continue to look for valid	Unknown problem—ignore; continue to look for valid sync
g) Invalid manchester	Error in message—ignore data in message	–
h) Improper number of data bits and parity	Error in message—ignore data in message	–
i) Discontinuity of data words	Error in message—ignore data in message	–
j) No status word response	Unknown problem—requires further investigation	Unknown problem—requires further investigation to achieve error determination

Table 5.1-4. Typical Error-Correction Techniques

Error identification types	Error correction technique
1. Bus system failures a) No status word response b) Message error c) Parity error d) Invalid manchester e) Improper number of data bits and parity f) Discontinuity of data words	• Retry message on same bus n times • Retry message on alternative bus n times • Transmit status word mode code on each bus If necessary, transmit initiate self-test mode code • Transmit BITE mode code • Analyze failure and determine corrective action, which may involve the following mode code commands: • Shut down transmitter (00100 or 10100) • Inhibit terminal flag bit • Transmit reset remote terminal mode code
g) Busy	• Retry message on same bus after a fixed delay time
h) Terminal flag	• If necessary, transmit initiate self-test mode code • Transmit BITE mode code • Analyze failure and determine corrective action, which may involve the following mode code commands: • Shut down transmitter (00100 or 10100) • Inhibit terminal flag bit • Transmit reset remote terminal mode code
i) Improper sync	• Ignore and reset for valid sync
j) Subsystem flag	• Normal data communication messages (address/subaddress) to examine sensor BITE discretes or words
2. Sensor failure a) Discretes b) BITE data word(s)	• Analyze failure and determine system-oriented corrective action

The above discussion is a description of the common requirements of terminal hardware in bus controllers, bus monitors, and remote terminals. With respect to a remote terminal, 1553B says: "Any data words(s) associated with a valid received command that does not meet the criteria specified in 4.4.1.1 and 4.4.1.2 or an error in the data word count shall cause the remote terminal to set the message error bit in the status word to a logic one and suppress the transmission of the status word. If a message error has occurred, then the entire message shall be considered invalid." Notice that the requirement is that the entire received message be considered invalid. This message invalidation requirement may cause some systems like electrical multiplex (EMUX) a problem. Since the EMUX systems usually have bit-oriented data rather than word or multiple words (message) oriented data, errors in a word following the reception of good data will invalidate good data. It has been proposed that such a system invalidate all data words from the failure to the end of the message and use previously good data words. This approach however, has not been allowed. Regardless of the approach, some system mechanisms will store the data and then tag the message as invalid. Others will not allow the user to receive the data. In the first case, it is the responsibility of the user to examine the message valid indication prior to using the data; however, in the second case, the user must recognize that the data has not been updated. What the above quote says, in effect, is that a remote terminal cannot use any part of a message that has an error. Message completion failures are always detected in a 1553 multiplex system and are known to the bus controller by either the suppression of the status word or the setting of the message error flag in the status word. This message error flag removes ambiguity as to whether the error occurred before the message was validated by the remote terminal or in the response to the message. Several points need to be made with respect to defining (or finding) the requirements imposed on software for message completion failures:

a. What indication will the bus control terminal hardware provide to the software that a transmission failure has occurred? Usually this is initiated by an interrupt, which causes software to examine hardware registers in the bus controller terminal.

b. What automatic retry of the last message does the bus controller terminal hardware implement, and to what extent is this under software control? See section 5.1.5 for a general description of the bus controller hardware interface to software.

c. What are the consequences of (1) ignoring the lack of message completion, (2) postponing action, (3) retransmitting the same message? This latter question may be important for the case when the message received at the RT was valid (and therefore used) and the message completion failure was caused by the origination, transmission, or reception of the status word.

d. What mode code usage has been planned for the avionic system? The 1553 mode codes provide a capability for investigating the details of a message completion failure. The mode code usage is optional, so the software designer needs to know, for each RT, what mode codes it is capable of decoding and using. It is desirable that each RT have the same capability of response to mode codes but, because of the availability of different types of hardware and GFE requirements, it is not always possible to do so.

5.1.4.2 Detected Subsystem or 1553 Terminal Failures

Subsystem or 1553 terminal failures may be detected using built-in-test circuitry. The 1553 standard makes provision for the reporting of either of these failures by the setting of the subsystem flag bit or the terminal flag bit in the status word to logic one. (The use of either of these bits is optional.) As part of the data output of a subsystem, the BIT or validity should be output for examination by the software using the data from that subsystem. Another method of determining multiplex system performance is by loop testing. Loop testing can be accomplished within a multiplex system at several levels. One method is for the bus interface hardware to examine its own transmission with its receiver and compare results to determine if transmission errors have occurred. Another method is to transmit a command to a remote terminal output circuit and monitor the output with an input circuit of the remote terminal and report the results to the software in the bus controller for comparison.

The requirements for action for this class of failures are more apparent than for message completion failures, since there is no ambiguity as to the type of failure, or its location. That is, given a detection of failure, what failure was detected, and what action should be taken? Again, several points need to be made with respect to defining the requirements imposed on software.

a. If dual-redundant buses are used, a terminal failure may be isolated to one bus. Depending on the capability of the remote terminal hardware and mode codes implemented, the transmit BIT word mode code can be a powerful diagnostic aid. Note that this mode code may not be used to request subsystem built-in-test results. The obvious implication for software is storing of the word format so as to determine the fault when the word is requested and received. For each fault, the action to be taken must also be determined, designed for implementation by software, coded, and tested.

b. Determining which subsystem failure caused the subsystem flag is more complex because there is no mode code similar to transmit BIT word for subsystems associated with a remote terminal. Polling of the subsystems connected to the terminal and evaluation of the responses may be required.

Subsystem or terminal failures can be detected without the use of the optional terminal or subsystem flags. For example, repeated message completion failures to a remote terminal via all possible data paths are likely to indicate the loss of function of the terminal. Bad data or nonvarying data from a subsystem may be interpreted as a subsystem failure. Software should be used to detect these and other failures. This subject is discussed in section 5.1.4.4.

5.1.4.3 Failure of Either the Bus Controller's Terminal or Processor

Catastrophic failures of the bus interface unit hardware or the processor (CPU, memory, I/O) can be grossly classified as total loss of function or incorrect and intermittent operation. A systems approach to the failure of the bus controller is the use of hardware discretes that simultaneously disable primary bus controller and enable backup bus controller. Except for

extremely unlikely failures, this type of mechanization avoids the ambiguity and bus crashes of two competing controllers. Therefore, recognition of failure of the primary bus control function (either the host processor or the bus interface unit) must be a requirement for a fail-safe enabling of a backup. Several methods, based in part on current implementations, are suggested here. All of these involve both hardware and software. These are--

a. The use of discretes and timers
b. Data bus monitoring by the backup controller
c. Use of the backup controller as an active element
d. Manual switch to backup by the flightcrew
e. A completely dual, triplex, or quad redundant system

The terminal fail-safe capability required by 1553B (par. 4.4.1.3) prevents incoherent constant transmission by the hardware. Since the timeout can be reset (see 1553B par. 4.4.1.3), the analysis of what sequence of actions is appropriate must be done carefully to avoid repetitive resetting of the terminal of the bus on which the timeout occurred. This case reduces to total loss of function, and hence silence, at the end of the timeout period, or "immediate" silence. A gray area is the intermittent failure of the bus controller but not complete failure of its bus control function.

In each case, the requirement must be to provide a verifiably fail-operational takeover by the backup controller (or alternative controllers) and to "keep it simple." In failure modes and effects analyses, generic failures of the software are very difficult to detect except for elementary-type failures. It behooves the software designer, therefore, to isolate the "switchover" software and to carefully restrict its access by other software.

5.1.4.4 Detected Data Errors by Software

The 1553 data bus does provide superior error detection capability for messages intended to be transmitted and received. This does not mean that inherent errors in data are also detected. Therefore, a software engineer should include data reasonableness checks or other authentication before data are used. This is a normal requirement, particularly for any data that are critical to mission success. Methods appropriate for transmission errors detected by software in 1553 data bus systems are--

a. Use of "tag" words. Recall that 1553 establishes stringent requirements on the terminal hardware design to detect errors in words, message continuity, or message word count. Also recall that if errors are detected in word count the message is not to be used. Since validation cannot be completed until the message is completed, hardware designers must make provision either for buffering and discarding an invalid message or tagging it as invalid. Tag words are generated by some hardware designs, and the tag word becomes an attachment to the received message. An elaboration of this idea is to also include the minor cycle number and the number of data words in tag word fields. The application software should examine the tag words on all data that it has received to determine whether the data are valid. The tag word is the only indicator of whether those data were transmitted properly and whether they were transmitted during the anticipated minor cycle.

b. Use of error detecting and correcting codes

c. Echo checks of data

d. Multiple copies of critical data items.

Techniques b, c, and d are related to data that users consider so important that the very small undetected bit error rate of 1553 systems is not tolerable.

5.1.5 Bus Interface Hardware Control

5.1.5.1 Types of Bus Interface Designs

In early 1553 data bus systems, some bus controllers were implemented that are not now in current use. An example of such would be a programmed hardware (PROM or ROM) controller, with no significant mission management capability. All current designs of bus controllers are of processor-coupled type. In this type of hardware configuration, the bus controller is linked to a bus control function. This bus control processor normally has additional "mission" processing requirements. The software complexity to implement this bus controller design is totally dependent on the sophistication of the bus controller hardware. That is, the more simple the design, the greater the software interaction required to process a command or message. Examples of bus controller capability are--

a. Single Word. This most primitive of processor-coupled bus controllers requires software interaction for every word of the message. Though a software routine can be written that specializes in transmitting words to the bus controller, the message processing burden remains in the software.

b. Single Message. This type of bus controller has the capability of processing one complete message at a time. The processor software sends the starting memory address of the message to the bus controller, which, in turn, performs a direct memory access (DMA) into the processor memory for the required message. The message, including the command word, is completely formatted by the software. The bus transaction is then processed under direct control of the hardware, which signals the processor at the end of message. The software is then left to examine the returned status information (status word, etc.) in order to ensure message completion. In this type of bus controller, the software interaction is lessened by the added capability within the hardware.

c. Multiple Message. This type of bus controller features hardware to allow processing of more than one message with a single software action. This means the bus interface hardware can recognize a normal message completion and all message completion failures. A set of messages is structured and their starting addresses are placed into a table in memory. To initiate processing of these messages, the starting address of the table of addresses is passed to the bus controller. The bus controller commences DMAs into the processor memory, stepping through the table of addresses all messages are processed, an interrupt is designated, or until an error occurs. An activity-complete signal and status word exceptions are returned to the software for examination,

just as in the single-message bus controller. Though the software is simplified by the added complexity in the hardware, considerable software activity may still be required for those applications where message structure or table organization must vary during the performance of the bus control function. The method by which the last message in the table is recognized is defined by the particular design. This type of controller is clearly the most desirable I/O controller for most applications.

5.1.5.2 Description of a Generalized Bus Controller Interface

The bus controller's bus interface unit is sometimes called the bus control unit (BCU), although the term bus interface unit (BIU) is also used. When this latter term is used, the context of the discussion is the key to determining whether the BIU is the interface for a remote terminal, bus controller, or bus monitor. Designers of BIU hardware frequently design the BIU as a BCU so that it operates in conjunction with a host processor. The operation of the BIU as an RT is similar to the operation of the BIU as a BCU. The following is a discussion of a generalized BCU processor interface and BCU operation.

BCU Processor Interface

A BCU is a distinct hardware item for 1553 data bus serial input-output. It therefore contains both the interface to the data bus (not discussed here) and the interface to the processor. It is usually a card or cards in a processor LRU, which functions as a data channel, and as such includes DMA control, program control, and interrupt control in addition to connection to the processor's internal 16-bit data lines.

Several points are relevant to the software designer:

a. Since the 1553 data channel has DMA control (top priority), the combined operations of the processor CPU and the 1553 data channel may simultaneously require DMA, effectively limiting the processor throughput. Therefore, throughput estimates should account for delay as a result of 1553 data channel activity.

b. The 1553 data channel will respond to software by the use of programmed input-output (PIO) instructions, and it is this feature that allows sequential and parallel operations of the data channel and other software.

c. The 1553 data channel can be interrupted by the processor, and the channel will interrupt the processor to signal completion of messages and errors. When a processor interrupt occurs, software must determine the cause of the interrupt.

The BCU contains its own internal registers with many design differences. In some cases, the channel control words (CCW) contain only the minimum information required, which is--

a. A pointer to the message to be transmitted (starting address, number of words)

b. A pointer to the address where the status word and received message will be stored

Basically, there seem to be two types of implementation of the channel control philosophy. One implementation uses the CCW to point to the 1553 command word and data words, and the software must format both the command word and the data words. Another implementation uses information in the CCW (e.g., the terminal address, number of words) to format the command word. In either case, however, the format of the CCW remains the responsibility of software.

BCU Operation

PIO commands (or memory-mapped I/O commands) are used to initialize and modify the BCU's internal registers. When a start command (another PIO) is received from the processor, the BCU begins transmitting. The BCU will continue processing channel control words until--

a. It recognizes a normal end of the CCWs.

b. It detects an error condition (viz, subsystem flag and terminal flag in the status word, and message completion failure).

c. It is halted by a PIO from the processor.

d. It is programmed to interrupt at the completion of a message.

e. The service request flag, set by an RT, causes an interrupt.

f. Busy bit, broadcast command receive bit, or dynamic bus control acceptance bit causes an interrupt.

Channel control words are also used by the BCU to receive instructions on the operation of the BCU. Software designers must know the formats of those words and their use by the BCU. Generally, specific fields in a 16-bit word (or words) are defined for the following operations:

a. Interrupt generation bits to establish when and if the processor is to be interrupted (For example, nonreceipt of a status word is a possible condition for interrupt.)

b. Error recording bits to establish the type of error detected by the BCU (For example, the BCU may detect parity error in a status word.)

c. Bus identification bits (or transmitter-receiver identification) to establish which of the redundant buses to use

d. No-operation bits to skip a bus message

e. Retry bits to establish the number of retries on a bus (if no status word was received) before interrupting the processor

f. Link bits to indicate that another word in a channel control block contains the address of the next CCW to be used

5.1.5.3 Considerations for Control of the Bus Interface

The requirements, then, for BIU control by software are fairly obvious:

a. Programs must have the capability to issue PIO commands to start and stop the BIU in sequence with other operations.

b. It is usually desirable to have the capability to format and modify the bus messages, or at least to skip (no operation) or jump (use link) to retain flexibility in operation.

c. It is a requirement to have an "interrupt handler" capability for interrupts caused by the BIU.

d. The specific BIU control details, CCW formats, etc., should be obtained as early in system development as possible.

e. It is good practice to isolate the BIU control software from other software.

f. The software designer should work closely with system designer who writes the hardware specification for the 1553 interface to ensure the software does not have to interface with hardware that routinely interrupts the processor after each message.

g. There should be no feature of the software that makes it specific to a given set of message lists, messages, or interpretation of received status words. In fact, since the format of status words varies somewhat from systems designed to 1553 (USAF), 1553A, and 1553B (as well as nonstandard formats), status word interpretation may need to be considered depending on the design.

In section 5.5, an example of bus control software is presented that should aid in understanding the points made above.

5.2 CONTROL OF APPLICATION SOFTWARE IN A MULTIPLEX SYSTEM

The key to the control of application software is the executive or supervisor. In avionic systems that have more than one processor on the bus, a multiplex system controller (MSC) or system executive is required to manage the avionic system. This is so because each "intelligent" device, such as an INS or backup or alternative bus controller, will usually contain software functions that are interdependent.

An executive in a multiplex system performs several distinct basic functions. These functions include bus control, other I/O device control, memory management, task sequencing, task-to-task communication, task scheduling, and time management. If tasks exist in more than one processor, these tasks also must be coordinated by minor cycle synchronization. Another function that can be thought of as being part of an executive but is possibly better considered an application task is that of the avionic system configuration control. This function is responsible for the error recording and handling for all the avionic equipment attached to the bus. In order to allow this function to be external to the executive, an application interface to the executive is required to allow communication with devices

to be changed via application requests. When an avionic device fails, an application task establishes communication with another redundant avionic device through an executive interface and stops or reconfigures avionic tasks that are dependent on data from the failed device. A similar requirement is necessary to schedule tasks by mission mode.

5.2.1 Duties of the Multiplex System Controller

The MSC software manages the multiplex elements in both normal and error conditions. Normal operating conditions include--

a. Initialization of the bus

b. Effecting the transmission of messages

c. Setting up of proper output message sequences

d. Timing for periodic message sequences

e. Aperiodic message setup, transmission, and/or reception

f. Communication of time event or message event (arrival) to an application task controller or to the recipient task

g. Transfer of control

h. Reconfiguration due to change of mission mode.

Abnormal operating conditions include:

a. Handling transmission errors

b. Responses to failure of subsystems on the bus

c. Failure recordkeeping

d. Reconfiguration because of failure

Each of these operating conditions is discussed below.

5.2.1.1 Normal Message Transmission

System functions performed by software are largely periodic and the processing is allocated to major cycles, each having two or more minor cycles.

The MSC software must initialize the BIU and direct it to the location of the first command (as in a channel control word) and the beginning of the message arrival and transmission areas, because it must perform both input and output operations. Once the BIU has been commanded to begin the transmission sequence, there should be no interaction with the BIU required of the MSC until the entire sequence of messages has been completed. At completion, the BIU may be directed (as part of a command) to interrupt the processor to notify the MSC that the transmission is complete. Once the transmission has been completed and the minor cycle interval is complete,

the MSC must switch the transmission list to the list appropriate to the next minor cycle. This list may contain some of the same message commands as in the previous minor cycle. Repeated messages are the most frequent messages that establish the selection of the periodicity of the minor cycle and the repetition of messages.

Aperiodic message setup is one of the most time consuming functions of the MSC and normally should be avoided if at all possible. Aperiodic messages can occur either from within the controlling computer or from any of the other remote terminals (processing elements) in a multiplex system. Aperiodic messages generated within the processor involve the following general steps: (1) the task creates a message to be transmitted, (2) the task communicates to the executive or MSC that it wishes the message transmitted, and (3) the MSC determines whether the BIU is transmitting or idle. If the BIU is idle or transmitting self-tests, the message is sent out by formatting a command and directing the BIU to read the command. If the BIU is not idle, the MSC must either add a transmit command to the transmission list so that the message will be transmitted at the end of the list or interrrupt the BIU once it has completed its current transmission. The MSC must then determine the command in the list that it was executing and link the newly created command to the next command to be executed.

If the aperiodic request occurs from a remote terminal rather than the bus controller processor, the following sequence of operations must occur: (1) the BIU must interrupt the bus control processor because the last message from the terminal had a status bit set (the service request bit) requesting an aperiodic transmission to occur, (2) the MSC must analyze the BIU registers to discern that an aperiodic request is present (rather than a message completion failure or any other status flag), (3) the MSC must initiate a message for the remote terminal (i.e., transmit vector word mode code) to transmit the identity of the aperiodic message that the RT intends to be transmitted if more than one aperiodic message is available from the remote terminal (obviously, if the remote terminal has a single aperiodic message, no transmit information need be requested since the bus controller would have prior knowledge of the subaddress and word count), and (4) the MSC must decode the aperiodic message request and add the appropriate message command to the BIU list so that the message may be transmitted . Even the last step may involve some interrupts if there is a requirement for an immediate response.

If a processing element is the recipient of an aperiodic transmission (sometimes indicated by reserving subaddress 30), the processor may be interrupted by the arrival of the message. If the processing element is interrupted, the executive must determine the cause of the interrupt and take the appropriate action relating to the message. The message could be related to the occurrence of an event that would require the initiation of one or more tasks.

A particular periodic message of importance is the minor cycle event notice. If more than one processing element is functioning using the same minor cycle sequence, the MSC normally transmits a minor cycle update or event to notify the remainder of the system that a new minor cycle is beginning and that message lists and pointers should be appropriately adjusted.

If processing has not completed by the expiration of a minor cycle, the software designer may elect to extend the minor cycle before going on or may begin a new minor cycle and assume that the processing will "catch up." The latter approach is very dangerous and must be examined carefully to determine whether the unfinished tasks of lower priority provide data for use by higher priority tasks. If such is the case, the sequence of tasks may become disastrously confused.

A function that may be necessary is the transfer of control from one processing element to another on a normal basis. This transfer requires handshaking prior to the transfer to ensure that the recipient processing element is operating validly, and posttransfer handshaking to ensure that the transfer has correctly occurred. Of special concerns are that the transfer would not occur correctly and that there may be either no bus control master or two bus control masters.

If several bus controllers are involved and a polling is the means by which a selection is made, the highest priority processing element could be offered the bus control (both the previous controller and a processor that is designated to monitor the transfers must receive notification of the beginning and ending of the transfer of control).

5.2.1.2 Abnormal Operations

5.2.1.2.1 Transmission Error

If an error is detected in a transmission, the BIU will either retry the message (some BIUs allow up to three retries on a message error) or interrupt the processor with notification of a transmission error. The MSC must determine whether to attempt to retransmit over the same bus or a redundant bus. The MSC should also retain records concerning errors for maintenance purposes and for terminal management. Retry philosophies vary, from a single retry on the redundant bus to retries on the same bus and retries on the alternate bus, according to the type of message being transmitted and the error handling capability of the bus controller. An example of error classifications is shown in table 5.2-1.

5.2.1.2.2 RT Failure

If a remote terminal fails, all transmissions to that RT address must be deleted from the transmission command lists. The message deletion is required to avoid repeated and time-consuming error detection and message retries. This deletion can be done a number of ways, depending on whether a processor is time or space limited. The easiest mechanism is to simply NO-OP the command lists for the desired RT address. A NO-OP is a BIU command that will cause the BIU hardware to skip this CCW. Similarly, subaddresses that are associated with a particular subsystem can be deleted. If response to such errors requires more speed of processing, pointer lists to each CCW using a particular address and subaddress can be constructed to access immediately each of the commands using that subaddress. If specific subsystems interface to multiple RTs or sensors are redundant, the MSC is responsible for the change of commands accessing one remote terminal or sensor to the remote terminals or sensors interfacing the redundant equipment. The message structure must have been carefully considered to enable such a transfer. If two terminals have identical

Table 5.2-1. Message Retry Procedures

Class retry	Purpose	Procedure	Applicable message operation
Class I retry (Auto retry)	Immediate retry of the bus message one, two, or three times.	BCU automatically retries bus message on same bus one, two, or three times (as specified in bus instruction) before presenting error interrupt to processor.	Most synchronous operations and most asynchronous operations not involving a remote processor.
Class II retry (careful retry)	Retry of the bus message only if the terminal had not successfully received the message. (Terminal required to have last command capability.)	Master processor obtains the last command and BIT registers via mode commands. If the last command is not the same as the bus message or an error is indicated, the bus message is retransmitted. Otherwise, the master continues with the next scheduled bus message.	Synchronous or asynchronous messages to the remote terminal/subsystem when repeated transmission of the same data will cause incorrect operation or degrade performance of the subsystem.
Class III retry (message sequence retry)	Retry of a bus message only after previous messages in the sequence have been repeated.	Master restarts the entire message sequence when error occurs with any message in the sequence.	Synchronous or asynchronous message to a subsystem when the entire message sequence must be received for correct operation of the subsystem.
Class IV retry (straight retry)	Confirm the remote did not receive the message properly and retry message after BCU has advanced instruction address register.	Analyze last command and BIT word as above, then reduce the value of the IAR by two and restart BCU.	Asynchronous messages from a transmitter (master or RT) that do need to be realigned.
Class V (realign transmitter)	Determine if the transmitter must be realigned prior to repeating the message.	Save the IAR minus 2. Obtain the transmitter's last command and status via mode code 18. If last command is correct, send a retransmit message to realign the remote transmitter. Restore the IAR and repeat the message.	Asynchronous messages from remote processor to master or an RT insensitive to repeated data.
Class VI (confirm and realign)	Confirm that message was not received properly, then determine if the transmitter must be realigned prior to repeating the message.	Save the IAR minus 2. Obtain the receiver's last command and BIT words via mode codes 18 and 19. If last command is correct and no message error indicated, continue with next bus message (i.e., do not retry). Otherwise, obtain the transmitter's last command and status via mode code 18. If last command is correct, send a retransmit message to realign the remote transmitter. Restore the IAR and repeat the message.	Asynchronous messages, remote to remote, or remote to RT when a careful retry is required by subsystem.

equipment attached as shown in figure 5.2-1, it is straightforward to translate the messages from one remote terminal to another (assuming that messages are not currently coming from the redundant RT). If the devices are cross-connected to different remote terminals as shown in figure 5.2-2, messages should be arranged in such a way that each message requiring remapping could be remapped without repacking of messages. If dissimilarly redundant equipment were to exist, the MSC would be responsible for the remapping of the messages to initiate the redundant communication with the secondary sensor (e.g., AHRS rather than INS data) and initiating the reconfigured software to process the new sensor type.

To avoid the problem of restructuring CCWs, it is better to have the entire set of messages transmitted each major cycle. The data from those messages do not necessarily have to be used by the application software simply because they are there. In this case, it is simpler to reconfigure following an equipment failure by eliminating that set of messages from the message command list and causing a different set of software to be initiated to process the different sensor characteristics (i.e., if the redundant sensors have differing characteristics).

5.2.1.2.3 Processor Failure

If a remote processor should fail, it is necessary to detect the failure and reconfigure the software to accommodate a different suite of processors to perform the computational functions. When considering the mechanism for control takeover from a failed bus controller, the best philosophy to follow is a "fail-passive" technique. Transmission from the failed controller BIU must be stopped and the failure must be noted. Some system architectures provide validity discretes between the master and potential backup system processors as a means to note the failure and to initiate recovery procedures.

5.2.2 Application Task Control

The purpose of this section is to define the relationship of the application software control to the use of 1553 as a communication medium between avionic equipment and one or more mission computers. Usually mission computers process sensor data and integrate the data to perform a specific mission. Considerations affecting the relationship fall into the following areas:

a. Tasks must be scheduled to execute after the data are obtained from a 1553 bus message.

b. Data from the 1553 I/O channel must be made available to the tasks and data from the tasks must be made available to the 1553 channel.

The above two functions establish the requirement for communication between a multiplex system controller hardware and software and application control software.

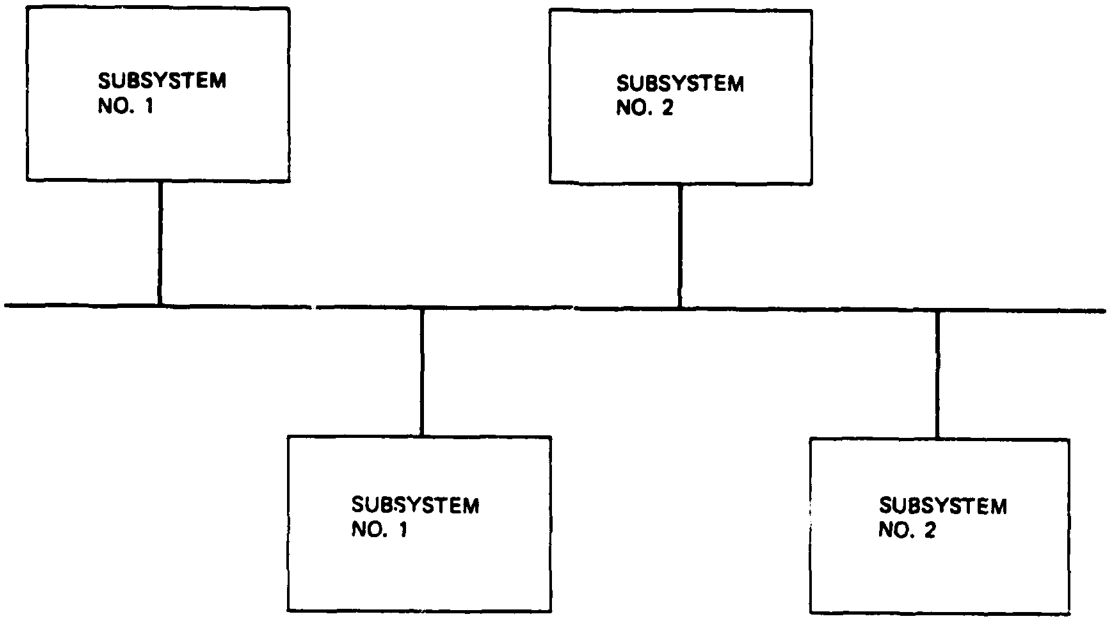


Figure 5.2-1. Redundant Equipment

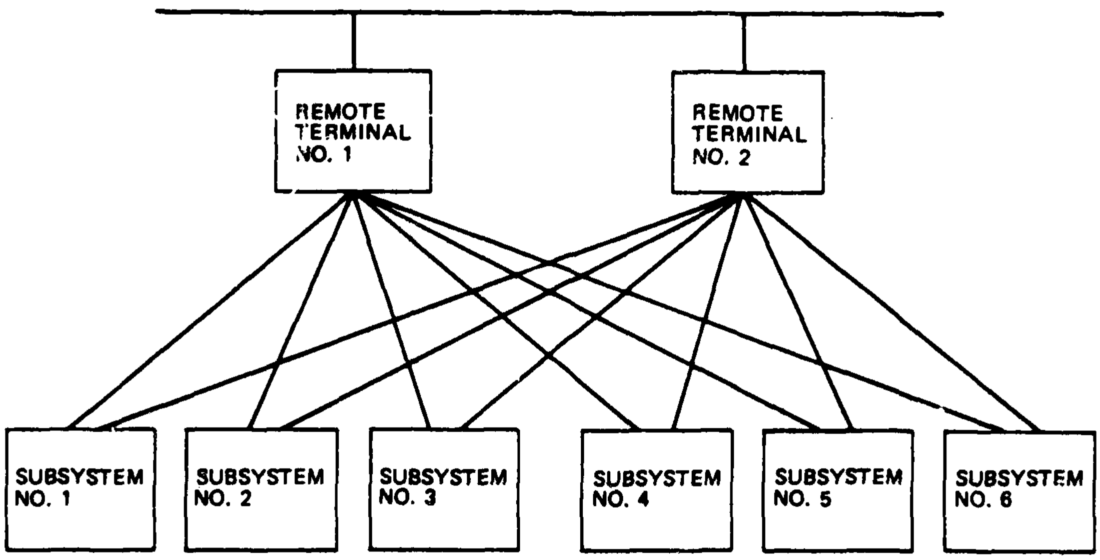


Figure 5.2-2. Cross-Strapped Equipment

a. If tasks are in multiple processors, a task in one processor may have a requirement to initiate the execution of a task in the other processor. This capability would require an executive-to-executive communication to cause the scheduling of a remote task or the communication of an event from one processor to the other.

b. A task in one processor may wait for a remote event, such as the arrival of aperiodic data in normal operation or the failure of some avionic device in abnormal operation. The executive may require the capability to interpret the arrival of a specific type of message (aperiodic) or a synchronize mode code as an event that could cause the execution of a waiting task. A minor cycle event may be transmitted by one processor to the other avionic equipment and processors to cause the scheduling and execution of tasks waiting for that particular minor cycle.

c. Messages may be requested to be transmitted at a particular time by application software. For example, the cargo release point or weapon delivery point may be predicted by the software for a specific time within a minor cycle. Because of accuracy requirements it may be important to transmit that message "when desired." The executive interface with the application software should then provide such a capability if required by the application.

5.3 MUX IMPACT ON APPLICATION SOFTWARE

The 1553 multiplex system is an architecture that allows all hardware elements attached to the bus as remote terminals or processing elements to operate in parallel. Normally, allocation of applications to processors is along functional lines, although high throughput processes should be distributed among processors. High throughput processes operating in parallel offer the advantage of increased system throughput (or decreased capability processors) if this does not cause overall system timing or bus loading problems. If more than one processor contains application software, these application processes must be constructed to accommodate parallel processing using 1553 as a medium of communication. The normal mechanism to accomplish this communication is by periodic messages with minor cycle synchronization. Within a minor cycle it is not easy to determine the progress of computation within any of the processing elements, except for aperiodic messages that will be discussed later. The normal mode of operation for application software in parallel processors to communicate is as follows:

a. Minor cycle 1: Process 1 creates data for output
b. Minor cycle 2: Data are transmitted via the bus from host 1 to host 2.
c. Minor cycle 3: Process 2 operates on data from process 1.

It is difficult to shorten this cycle for normal messages because inputs and outputs via 1553 occur as DMA operations concurrent with the operation of the processing elements executing the applications processes. It is difficult to determine when a particular message will arrive at a processor because of potential message retries and minor variations in processing time. Some data may require time tags. It is possible to determine when a particular message is transmitted since the BIU command (instruction) can cause an interrupt to the processor when the transmission is complete;

however, interrupting the processor is an extraordinary measure that should not be done with normal message processing.

A consequence of the three minor cycle communication principle is that separated communicating processes should plan for that amount of time to elapse between processes. The number of minor cycles is usually defined as a function of the number of sequential communications that must take place and the frequency of execution of the most rapidly repeated function.

5.3.1 Process Allocation and Construction

Since 1553 provides for the possibility of distributed processing, one of the significant aspects of software design is the decomposition and allocation of processes to processors. Processes may be potentially reassigned from one processor to another if problems in either timing or sizing arise during development or changes occur in the application or redundancy requirements. If any possibility of reassignment exists, a higher level of independence of these application functions from the remaining functions must be established. The other functions must be able to compute, and the mobile function must be able to receive and send data over the bus as a primary way of communicating with all other functions. One way to achieve this isolation of functions is to require that each task communicate with another only via partitioned data bases (e.g., J73 compools) and that the executive must be responsible for equating the data (or copies of the data) with the communication areas.

Another mechanism to use in the distribution of tasks is to break the functions into very small tasks that can be considered to be relatively independent and code these tasks. Once the entire system has been constructed and integrated, these tasks can be combined into larger, more efficient units called macrotasks. These macrotasks are invoked by the executive and the tasks are invoked by the control logic within the macrotasks. The ideal solution would be to retain all tasks at the finest granularity, invoked by the executive and communicating only through data controlled by the executive.

5.3.2 Error Tolerant Software

The communications between processors and remote terminals provide sources of errors that are avoided in a single processor system. Consequently, arriving data need to be carefully checked for validity in multiprocessor systems, because they could either be transmitted in error or they could be out of sequence. Out-of-sequence problems occur when one processor is one cycle ahead of another.

The BIU normally is responsible for examining a message for transmission errors (word validation, continuity, length) and depositing a tag word with the message to indicate the validity of the message. Some BIUs also indicate the minor cycle in which the BIU is operating. The BIU may create errors of its own and not indicate that a message is in error, so it is incumbent upon the software to validate the data. Certainly the tag word must be checked to determine the validity of the data and the age of the

data. By age is meant the minor cycle in which the data were delivered to the processing element. This check is useful to determine if newer data replaces the old data. If such is the case, some alternative processing by the recipient task is required to accommodate the situation.

Double buffering of data should be considered. Because of the possibility of the input data being destroyed by an error condition (note that the message is written into memory word by word and message validity is determined at the end of the sequence), the original data should be saved if it is necessary to use the data again. A convenient technique is to use alternating buffers on even and odd minor cycles, so that data are overwritten in a buffer except on alternate minor cycles.

5.3.3 Communication

Message packing is an important part of the data base design. It is essential that the messages not be full 32-word messages for greatest "efficiency." Just as the programs grow in size, so does the amount of data that must be communicated, especially between processing elements. It is recommended that the messages reflect functional output when possible and not be packed to more than 10 to 15 words per message, to allow for growth and change. The smaller the messages, the less potential impact on the remainder of the data base. During design, the data buffers may be allocated to be 33 words long (32 words plus tag word) to accommodate growth as the system communication requirements grow. This preallocation would have the least impact on memory allocation as the design progresses.

5.4 SUPPORT SOFTWARE

Support software that is specific to the 1553 system lies in the area of message transmission manipulation and task and multiplex interface manipulations. Software managers should consider preparation of special support software to aid in multiplex software development and to obtain increases in flexibility and productivity.

5.4.1 Message Manipulations Support Software

The BIU interface with the processor contains a number of elements that may require rearranging during implementation, testing, and integration. With each message the following characteristics could be expressed in terms of mnemonics and human readable fields and could be formed into a sequence of BIU instructions and memory locations: (1) message source, (2) message destination, (3) message length, (4) frequency of transmissions, (5) period to begin transmission, (6) primary bus on which to be transmitted, (7) message memory address, (8) subaddress correspondence, (9) number of times to retry the message if transmission error occurs, and (10) error routine to be called if the transmission cannot be completed.

These basic message characteristics can be used to create the processor and RT message-to-BIU interface. The tables that could be created with special support software include (1) the BIU command list for each minor cycle and the interconnection between command lists when appropriate, (2) the pointer list to the message memory locations (if such a pointer list is necessary for the BIU), and (3) the length allocated to each message.

5.4.2 Task-to-BIU Interface

The task structure is impacted by the multiplex communication structure in three areas:

a. The messages that arrive must be communicated to the tasks or the tasks must be given access to the message areas. If the message data are physically moved to task work areas, the linkage to perform the transfer and the allocation of the data area could be performed by the MUX support software.

b. The task must be mapped to the minor cycle in which it is to be executed. The task is normally tied to arrival of data over the 1553 bus and to minor cycle events. The support software could be designed to determine that one or both of the conditions are met and link the task to the sequence of tasks that are to be invoked in a given minor cycle.

c. Tasks must be allocated to processors if more than one processor is capable of performing them and if uncertainty exists during development of which processor may be allocated tasks.

5.5 BUS CONTROL SOFTWARE EXAMPLE

This section discusses the operation of a typical bus control unit both during normal operation and in the event of an error. For purposes of the example, dual-redundant buses operating in the active-standby mode are assumed. Only the primary bus controller and the information exchange of it with three remote terminals are included, using all of the 1553 information transfer formats. Prior to the discussion of the software, the example hardware interface must be described. A general description is given below and a more detailed description is given in section 5.6.

Each BIU is essentially an independent I/O channel using a common DMA interface. Each BIU has its own set of registers that interface with the host processor. During initialization the following registers must be loaded using PIO for each BIU command: the program control register (PCR), the instruction address register (IAR), and the base address register (BAR). Once the PCR is initialized and set to "GO," the autonomous operation of the data channel will continue until an error is detected or a programmed halt takes place. The overall sequence of operation follows:

a. The BIU obtains instruction word 1 (IW1) from the host processor memory via DMA. If IW1 indicates that the BIU controlling the other bus should transmit this particular message, control and I/O are passed to the standby BIU. IW2 is subsequently read from the processor memory.

b. The BIU generates the command word (or command words for an RT-to-RT) for the message using the information in IW1 and IW2.

c. The BIU generates the specific address of the message pointer in the message pointer block, which is the address of the message to be transmitted or the address of the data area for the message to be received, by using the BAR contents (10 bits) plus the T/R bit (1 bit) plus the subaddress (5 bits) in IW1 or IW2 (depending on the type of transmission).

d. The BIU acquires the message address via DMA from the message pointer block.

e. The DMA controller, in conjunction with the BIU, uses this address and the word count to transfer the data to or from the host processor's memory.

f. The BIU transmits the command word, and data words if any, and waits for the status word, and data words if any, to be received.

g. If data are to be received, the BIU writes the tag word that indicates the time tag from the master function register into the first word of the message area, the validity of the data input into the message area, and the data word count.

See figure 5.5-1. Additional details of the use of the registers and operation are given in the following sections.

5.5.1 Processor Control Register

The PCR must be initialized, using PIO commands, to indicate to the BIU the BIU address, which functions as a 1553 terminal address (TERMINAL ADDRESS); whether the BIU is enabled to run (GO/NO GO); whether the BIU is to indicate that the unit is busy (BUSY); the identification of the unit in terms of its control of bus 0 or bus 1 (BUS ID); whether the processor should be interrupted for mode data, asynchronous message requests, and master function-related requests (INHIBIT INTERRUPT); whether the BIU is to operate as a bus controller or as a remote terminal (MASTER/REMOTE); whether bus control can be accepted via a mode code (BUS CONTROL ACCEPTABLE); and other data concerning the internal loading and communication formats.

The PCR, once initialized, starts the autonomous operation of the data channel, which will continue until--

a. An interrupt is indicated by bit 6 of IW1 (see fig. 5.5-2).

b. A halt is indicated by an OP code in bits 1 and 2 of IW1.

c. The number of allowable retries is exhausted because of one or more message completion failures (see bits 3 and 4, IW1).

d. Bits in the status word received indicate an interrupt.

e. Built-in-test (BIT) determines a fault condition that requires an interrupt.

f. A halt is received from the host processor.

g. IW1 indicates that the command word is for the other BIU.

5.5.2 Instruction Address Register

The BIU IAR is a 16-bit address, which is incremented each time a BIU instruction word is obtained via DMA. For each pair of instruction words,

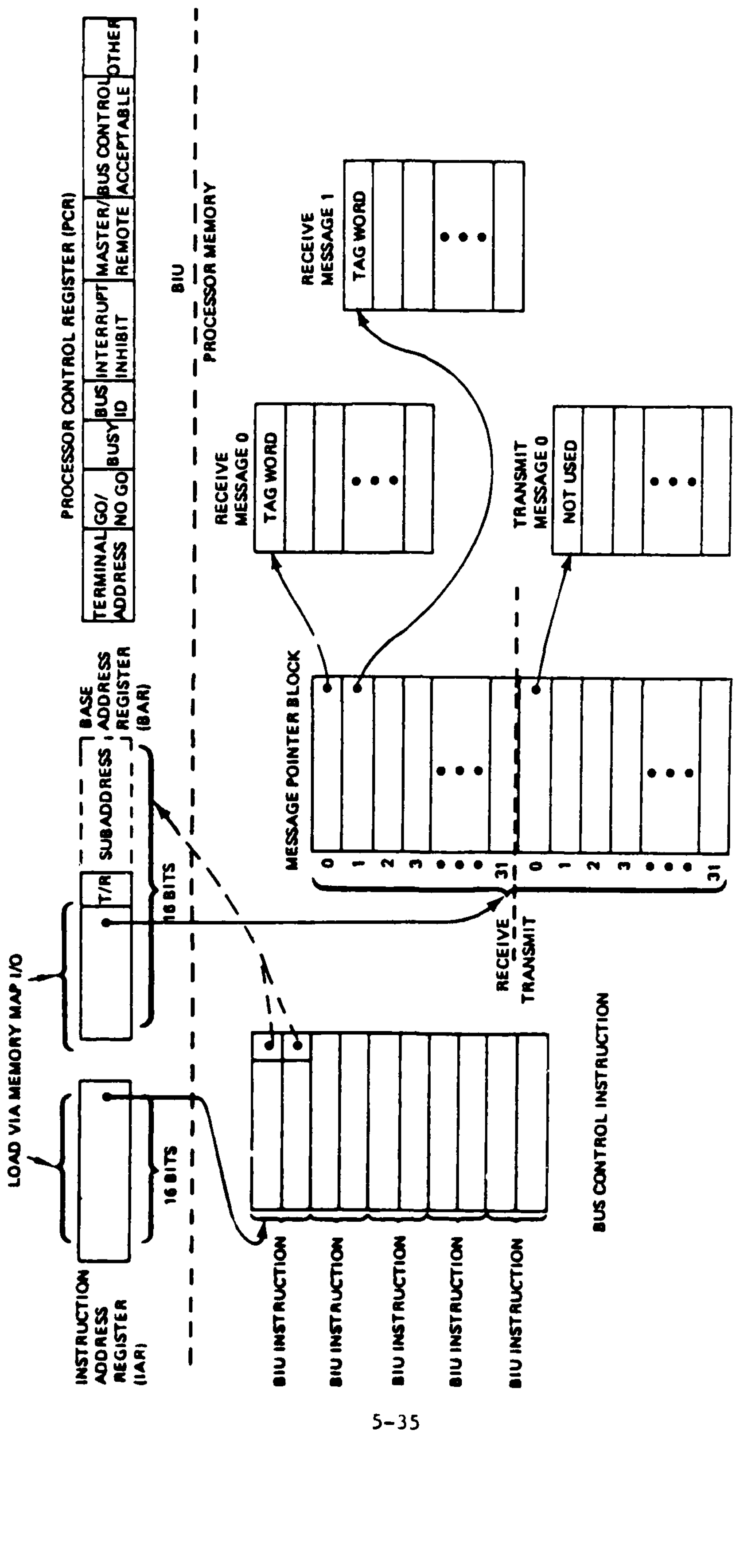


Figure 5.5-1. BIU/Processor Interface

the IAR functions in exactly the same manner as the instruction counter for a central processor, which points to the next instruction to be executed.

5.5.3 BIU Instruction Words

The BIU instruction word pair formats are shown in figure 5.5-2. They must be formatted by software. The instruction word pairs are used by the BIU for--

a. Constructing the command word or words

b. Determining which bus is to be used for transmission and reception, the mode of operation, the retry count, the interrupt condition, and the select condition

c. Constructing the address within the message pointer block of the message pointer

The source of information for the command word or words is the information in IW1 and IW2 data fields, except for the T/R bit. Since IW1 always contains the receive address and IW2 contains the transmit address, the BIU can determine how to set the T/R bit by comparison of its own address (contained in the PCR) with the terminal addresses in IW1 and IW2, according to the rules in table 5.5-1.

Provision has been made for BIU OP codes (see fig. 5.5-2) in IW1 so that an instruction pair may be skipped, (i.e., no operation) and the IAR will be incremented twice. The OP code can indicate that IW2 is the address of the next BIU instruction pair, allowing for jumping to instruction pairs out of sequence in memory by loading the contents of IW2 into the IAR. In addition to this flexibility, the BAR may also be changed to establish unique message pointer blocks, usually for transmitting and receiving different data during different minor cycles.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

BITS

NORMAL OPERATION	RETRY NUMBER	WHICH BUS	INTERRUPT WHEN COMPLETE	RECEIVING DEVICE NUMBER	RECEIVING DEVICE SUBADDRESS
3	3	0	0	9	10

WORD COUNT	SELECT BIT	TRANSMITTER ADDRESS	TRANSMITTING SUBADDRESS
7	1	6	13

Figure 5.5-2. Channel Control Word

5.5.4 Mode Data Register

If the BIU instruction word contains the mode identification in the subaddress field, and the mode code in the word count field is a mode code with data word, the MDR must be loaded with the data word to be transmitted. The same MDR is used to store any data word received as the result of a mode code sent by a remote terminal.

Table 5.5-1. BIU Comparisons To Determine T/R Bit

1553 message type	Receive device field	Transmit device field	T/R bit
RT to RT	RT address	RT address	0 for first word; 1 for second word
RT to bus controller	BIU address	RT address	1
Bus controller to RT	RT address	BIU address	0

5.5.5 Internal Status Register

The ISR, in combination with the BIT register, gives the normal and abnormal status information pertaining to the operation of the BIU. These indicators include BIU failed, program-controlled interrupt, invalid instruction, mode data present, asynchronous message interrupt, minor cycle interrupt, status word abnormal conditions, and fatal DMA error.

5.5.6 Status Word Data Register

A register in the BIU stores the status word from a receive command, and another register holds the status word from a transmit command. Each register is accessible by PIO command (or a memory command).

5.5.7 Built-In-Test Register

A register in the BIU stores the results of the BIU built-in test.

5.5.8 BIU to Processor Control Codes

The BIU will have a unique hardware interface to the host processor. Loading and outputting of the BIU registers is accomplished using a 4-bit control code. When interrupts and abnormal halts occur, the processor software will evaluate the indications in the BIU internal status register

and the BIT register to determine the next action. Interrupts normally will require the examination of both of these registers to determine the interrupt cause.

5.5.9 A BIU Scenario

This example starts with the setting of the necessary registers to initialize the BIU and to initiate the transfer of information from the bus controller. The first three messages are to synchronize the multiplex terminals and to determine that two specific terminals did receive the synchronize mode code. (If the system only consists of a bus controller and two terminals, the broadcast synchronize may not be either efficient or necessary.) Two cases are presented: one in which the broadcast of the synchronize operation occurred correctly and one in which a remote terminal failed to receive the command. Note that using a broadcast does not elicit a status response from the remote terminals and if knowledge is required that a message was received, then any terminal can be polled (transmit last command mode code) to determine if it received that broadcasted message as its last command. Consequently the first message of a minor cycle is a broadcasted synchronize mode code to synchronize the entire multiplex system, and the next two messages specifically poll two remote terminals and interrupt the processor to examine the results of each polling. The processor then sets the BIU to process the minor cycle message list. Each of the messages is transmitted in turn, and the last message halts the BIU and causes an interrupt of the processor.

The processor then resets the BIU to accept a new message list for the next minor cycle, determines that the broadcast was received and continues the processing. During this cycle an error in transmission occurs and the alternate bus is tried and is successful. An asynchronous message request is also received and executed in this cycle.

The BIUs are initialized using the following sequence of operations:

a. The BAR in each BIU is set using control code 6.

b. The IAR in each BIU is set to 2000 using control code 5.

c. The master function register is set with the first minor cycle.

d. The PCR is set in each BIU. One PCR will indicate that the BIU is for bus A and the other PCR will indicate bus B. Both PCR's will indicate that the controller number is 6, it is a bus controller, it is not busy, and it should function normally (GO bit true).

The bus controller commands are in contiguous memory locations beginning at location 2000. The BIU instruction address register (IAR 16 bits) points to the next command to be executed in the memory in exactly the same way that the instruction counter for the central processor points to the next instruction to be executed. The instruction address register must be loaded with a PIO (or memory mapped) instruction and can be read with another PIO instruction to determine the location of the next bus command. The IAR should only be read when the BIU is halted, because this register is changed independent of the operation of the processor. Commands shown in figure 5.5-3 are as follows:

a. Broadcast synchronize on bus. This command must be preceded by loading of the minor cycle number (or synchronizing number) from the master function register into the BIUs mode data register.

b. Request last command from remote terminal 7, load last command into mode data register when received, and interrupt on completion.

c. Request last command from remote terminal 8, load last command into mode data register when received, and interrupt on completion.

d. Transmit to remote terminal 9, subaddress 10 on bus 0 (A) using data message 13, which is seven words long. If a transmission error occurs, retry up to three times. Subaddress 10 is used by remote terminal 9 as the data reception address or pointer. Message 13 is the controller's index into the message pointer block, which contains the address (minus one because of tag words) of the first word of message 1. (Refer to fig. 5.5-1.)

e. Remote terminal 7 transmits an eight-word message from subaddress 4 to the bus controller to be deposited in receiver message 5.

f. Remote terminal 7 transmits a four-word message from subaddress 5 to remote terminal 9, subaddress 2. The message is not to be repeated if a failure occurs.

g. Halt the BIU normally and interrupt the processor.

Details of the channel control block execution sequence (fig. 5.5-3) will be discussed here. Once the master function register has been loaded with the tag word minor cycle indicator and the mode register has been loaded with the minor cycle number, the BIU may execute the first instruction. The GO bit is set and the first command is executed. This command broadcasts the synchronize mode command with a data word containing the time tag (minor cycle number) "1" as shown in figure 5.5-3.

The BIU instruction at location 2002 causes a mode code (18) to be transmitted to RT 7, which commands its last received command to be transmitted as a data word. The data word is sent to the BIU's mode data register and the BIU interrupts the processor. The processor must examine the BIU ISR to determine that the interrupt was programmatically generated and that mode data are present (see bit 4 and bit 6 of ISR in sec. 5.6.3.2). The processor must read the MDR to determine that the synchronize mode code was received. The GO bit must be reset to 1 in order to start the BIU at the next transmit last command mode code to be transmitted to device 8. The same steps are followed by the processor to examine the last command. Suppose that the "last command" returned was not the broadcast command, indicating that for some reason the command was not received by RT 8. Assume that it is permissible in this system organization to resynchronize by transmitting a synchronize to that specific remote terminal (rather than resynchronizing the entire multiplex system). Then the following set of instructions can be constructed to handle this error condition as shown in figure 5.5-4. The method of retry will be to transmit on the other BIU (bus 1) and then interrupt after interrogating the last command again, using the steps outlined above, assuming that this time the synchronize command was received. Then the final instruction to be executed

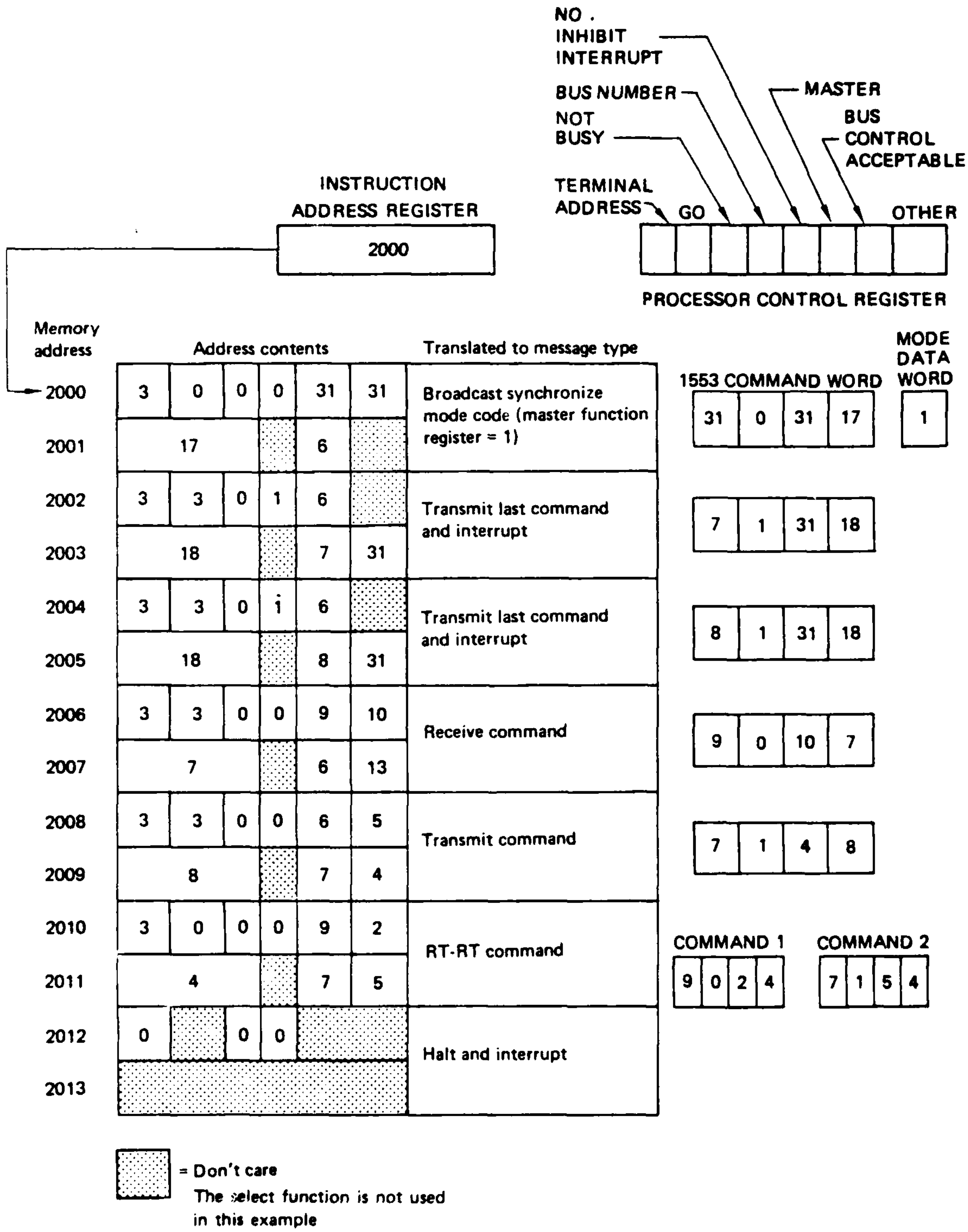


Figure 5.5-3. Channel Control Block for Cycle Number 1

3010	3	0	1	0	8	31	Synchronize mode code to RT8 on bus 1
3011	17				6		
3012	3	0	1	1	6		Transmit last command and interrupt
3013	18				8	31	
3014	1		1	0			Link to bus command at location 2006
3015	2006						

(shaded) = Don't care
The select function is not used in this example

Figure 5.5-4. Error-Handling Command Sequence

by the BIU from this set of instructions is a link back to the normal instruction sequence (location 2006).

The command pair at location 2006 causes RT 9, subaddress 10, to receive a seven-word message from the bus controller. The particular message was located by the BIU by indexing 13 locations into the receive portion of the message pointer block to determine the address of the particular message to be transmitted. The space allocated for this message, however, is seven words long.

The command pair at location 2008 commands RT 7 to transmit an eight-word message corresponding to subaddress 4. The message is deposited into message location number 5. The first word of that message area is the tag word, which gives the minor cycle number (1), the word count, and whether the data are valid.

The final terminal command is an RT-to-RT transfer. Notice that the master's terminal number 6 appears as neither the receiving terminal number nor the transmitting terminal address. Therefore, the BIU determines that an RT 7 to RT 9 set of commands, consisting of two command words, must be developed and two status words must be monitored for errors or incomplete message transmissions. The data associated with an RT-to-RT transfer may be received by the controller if the appropriate bit is set in PCR.

The final instruction of the list is to gracefully halt the BIU, which interrupts the processor. The processor must determine by examining the ISR that the sequence is complete. The executive must wait until the minor cycle time is completed and then start a new message list, unless an identical list of commands is used for each cycle. In this case, when the next minor cycle is initialized, the same message list is started. Double buffering of message data is usually required, because data are received each minor cycle and the software normally requires an entire minor cycle to operate on the data before the data are changed. If a single buffer were

used, the data might be altered by the BIU as the software is concurrently operating on it. As an example of a problem that could occur, assume that a two-word floating point number (mantissa and characteristic) is transmitted on the bus and is used by the software without buffering. Unless timing is strictly controlled, the software could access one word (the characteristic), which was updated by the BIU, and access the next word (the mantissa), which was not updated by the BIU. Therefore, the buffer should normally be switched from one minor cycle to another. This can easily be done by switching the message pointer block by resetting the base address register to the beginning of a new message pointer block. Switching the list of bus commands requires resetting the instruction address register to the beginning of a new list of commands. If there are 64 minor cycles, there could be up to 64 different message lists.

Assume that on the next minor cycle the message sequence is as shown in figure 5.5-5. The first three commands are the same as in all minor cycles: broadcast the synchronization and check two of the remote terminals. The time tags will now reflect minor cycle 2. The next command is also the same for every minor cycle. Probably at least one transmission will be the same for each minor cycle since the number of minor cycles per second reflects the highest frequency of data updates required for system operation. The next message transmitted is a 32-word broadcast to subaddress 2 from message 4 in the bus controller's memory. Then an RT-to-RT transfer between terminals 7 and 8 occurs. A transmit command causes 12 words from remote terminal 7, subaddress 18, to be moved to the bus controller's message 5. Finally, the BIU is halted and the BIU interrupts the processor to await further instructions.

Suppose that on the RT-to-RT transfer RT 8 has an asynchronous transmission that must be sent, consisting of a two-word message 17 to the bus controller. RT 8 will have set its status register with a service request bit. The status response, when received, will cause an interrupt by the bus controller. The bus controller must determine the type of interrupt (internal status register) and will discover that it is a service request. Next the address of the remote terminal must be retrieved from the status word and the transmit vector mode code transmitted to that terminal if more than one asynchronous transmission is possible from the terminal (see fig. 5.5-6). The internal status register will indicate whether the receive command status word or transmit command status word caused the interrupt. Once the vector word has been requested and read from the mode data register, the vector can be used to determine the specific message desired to be transmitted by that device. In the example, RT 8 has requested that subaddress 14 be transmitted in a two-word message. Once the sequence of instructions is complete, the link is made back to the normal sequence of instructions at location 2052.

The final example is concerned with the failure of a remote terminal. Assume that RT 9 does respond with a status word indicating remote terminal failure from the command at location 2046 (see fig. 5.5-5). That status word will interrupt the master processor and the master must determine through the internal status register, the BIT register, and status register the cause of the interrupt. These registers will indicate a failure has been set for RT 9. The status register must also be examined to determine if a terminal failure flag has been set. Once these three PIO operations have been executed and the data evaluated, that command in the command list

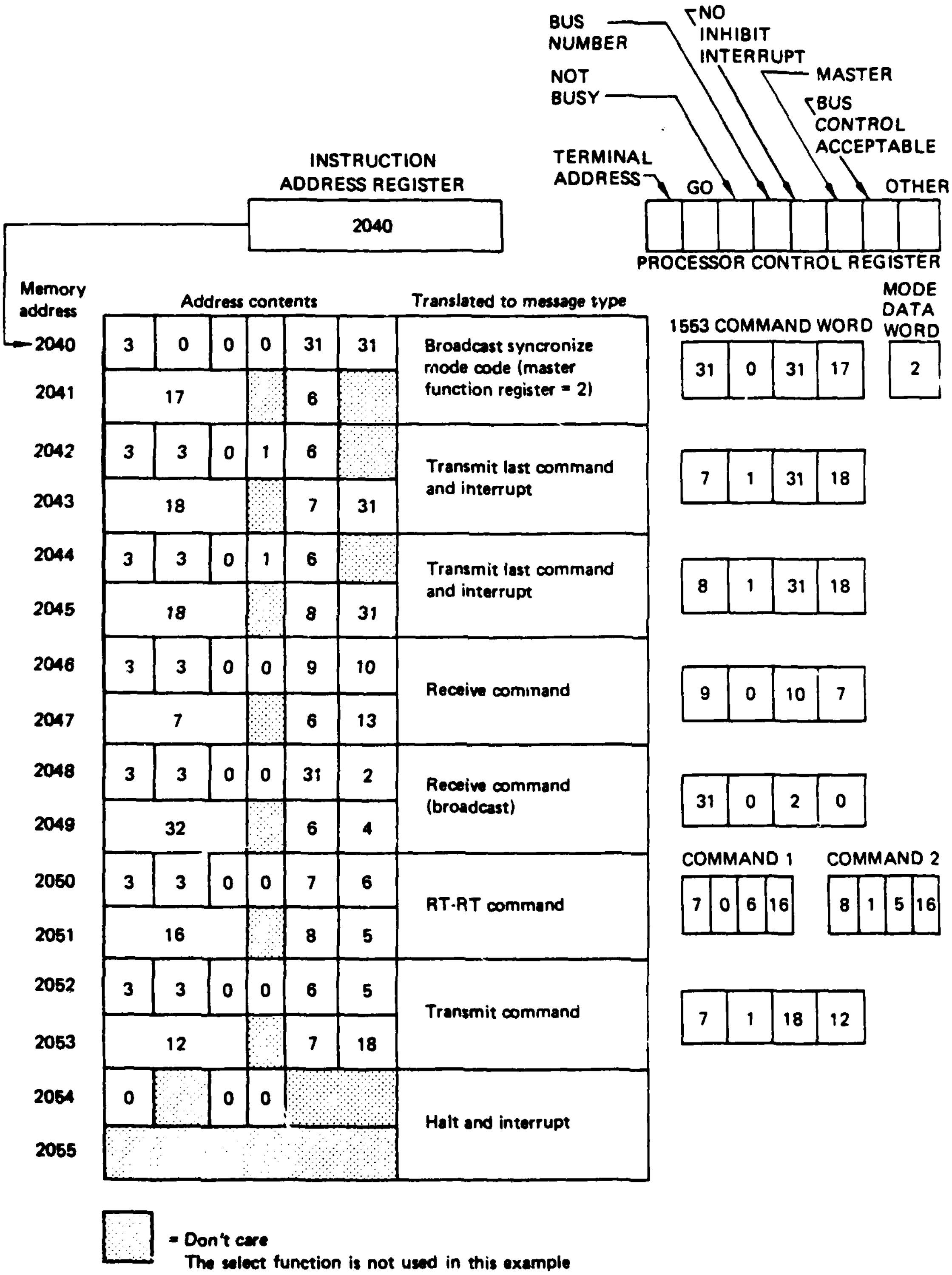


Figure 5.5-5. Example of Minor Cycle Number 2

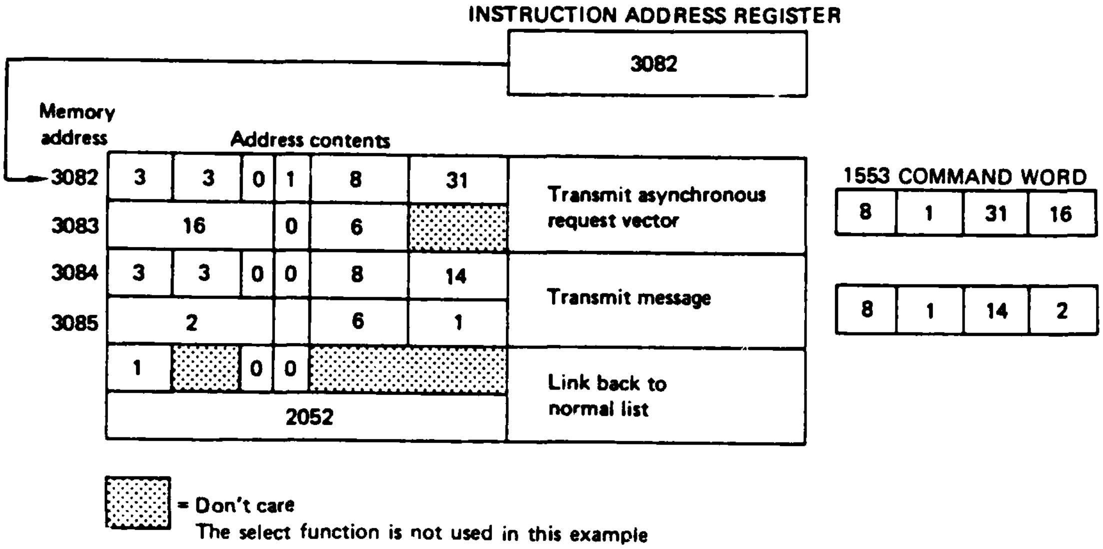


Figure 5.5-6. Asynchronous Message Transmission

will be NO-OP'ed, and the BIT from the remote terminal will be requested for further analysis by the error handling routines. The NO-OP operation involves the alteration of all locations involving RT 9. For example, in the BIU command in figure 5.5-7, the first field of location 2946 will be altered so as to indicate to the BIU to skip the command.

When (and if) the remote terminal recovers, that command can be included in the command list by changing the NO-OP (2) to a normal bus command (3). The implications of the failure of a part of the multiplex system should also be noted.

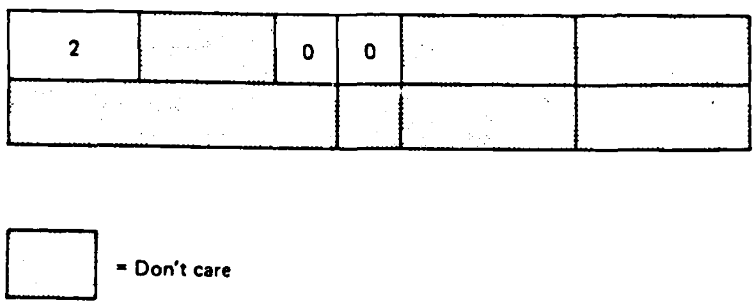


Figure 5.5-7. NO-OP Command

5.6 PROCESSOR TO BUS INTERFACE CHARACTERISTICS ASSUMED FOR THE EXAMPLE

This section describes in considerable detail the hardware interface of the software example discussed in section 5.5. The amount of detail included in section 5.6 is approximately the amount of information required to understand well the example given in section 5.5. This section describes the functions of the BIU primarily from the perspective of a bus controller and discusses the operation briefly as a bus interface for a processor acting as a remote terminal.

A key point to keep in mind is that software must control and interpret the entire interface as described here. Error determination and handling must be based on the specific BIU interface that exists for a given piece of hardware. Each BIU operates as an independent I/O channel in contrast to some BIUs that share common receivers and transmitters. In this implementation, assume that one BIU controls bus A and the other BIU controls bus B. Each has its own set of registers that are described below.

5.6.1 Stored Program Instruction Interface

During initialization, the BIU is given an address for its instruction address register, which points the BIU to its BIU instructions. Instructions are arranged in pairs that are stored sequentially in memory. The format for these instructions is given in table 5.6-1. The BIU is also given a base address register, which is 10 bits long.

Using this and the instruction words, the BIU can develop the address into the pointer block and find the data buffer. The first of a pair of DMA sequences occurs, the first instruction word is acquired, and the address register is incremented. The BIU initiates a second DMA cycle and acquires the second instruction word. The IAR is incremented once again to prepare for the next fetch operation.

Once the two instruction words are acquired, the BIU can construct the command word(s). Referring to table 5.6-1, the BIU compares its terminal address (available from the PCR control word given to it when initialized by the host, see fig. 5.6-1) with the device addresses in the instruction words. If the BIU's address is the same as the receive device address, the command to be generated is a transmit command to an RT. If the BIU's address is the same as the transmit device address, the command to be generated is a receive command to an RT. If the BIU's address does not agree with either device address, an RT-to-RT pair of commands is to be generated. As part of BIT, the BIU checks to ensure that the receive device address is different from the transmit device address.

When an RT-to-RT set of command words is formed, no data buffer address is generated, because the controller does not transfer data in that case unless the RT-to-RT data enable is set in the PCR (see fig. 5.6-1). When an RT is to receive data, an RT transmit command is generated and the BIU generates a corresponding data buffer address. As mentioned above, the BIU is given a BAR word (discussed later), and the BIU appends 6 bits (the T/R bit and subaddress bits) of the command word to the least significant end of the BAR word to form an address into a pointer table. The pointer, acquired from the table, points to the first address in the data buffer and is stored in the pointer register of the BIU. This first address is reserved for the tag

Table 5.6-1. BIU Instruction Format

<table>
<tr><td colspan="15"></td><td>LSB</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td></tr>
<tr><td colspan="4">OP code retry</td><td>B</td><td>I</td><td colspan="5">Receive device address</td><td colspan="5">Receive subaddress/mode</td></tr>
<tr><td colspan="5">Word count/ mode code</td><td>S</td><td colspan="5">Transmit device address</td><td colspan="5">Transmit subaddress/mode</td></tr>
</table>

IW1 bit designation

1-2 Normal OP codes
 Bit 1-2
 00 = Halt BIU
 01 = Link (use second word as address of next two-word instruction)
 10 = No operation (go to next two-word instruction)
 11 = Normal bus operation

3-4 Indicates number of retries (0,1,2, or 3)

5 0 = Operation is performed on bus A
 1 = Operation is performed on bus B

6 1 = Interrupt processor upon successful bus operation

7-11 Receive terminal addresses

12-16 00000, 11111 = mode command operation
 00001
 –
 –
 11101 } Receive terminal subaddress
 11110 = asynchronous message

IW2 bit designation

1-5 Word count or mode command code

6 Select
 Select bit 0 = Select output = 0
 Select bit 1 = Select output = 1

7-11 Transmit terminal addresses

12-16 00000, 11111 = mode command operation
 00001
 –
 –
 11101 } Transmit terminal subaddress
 11110 = asynchronous message

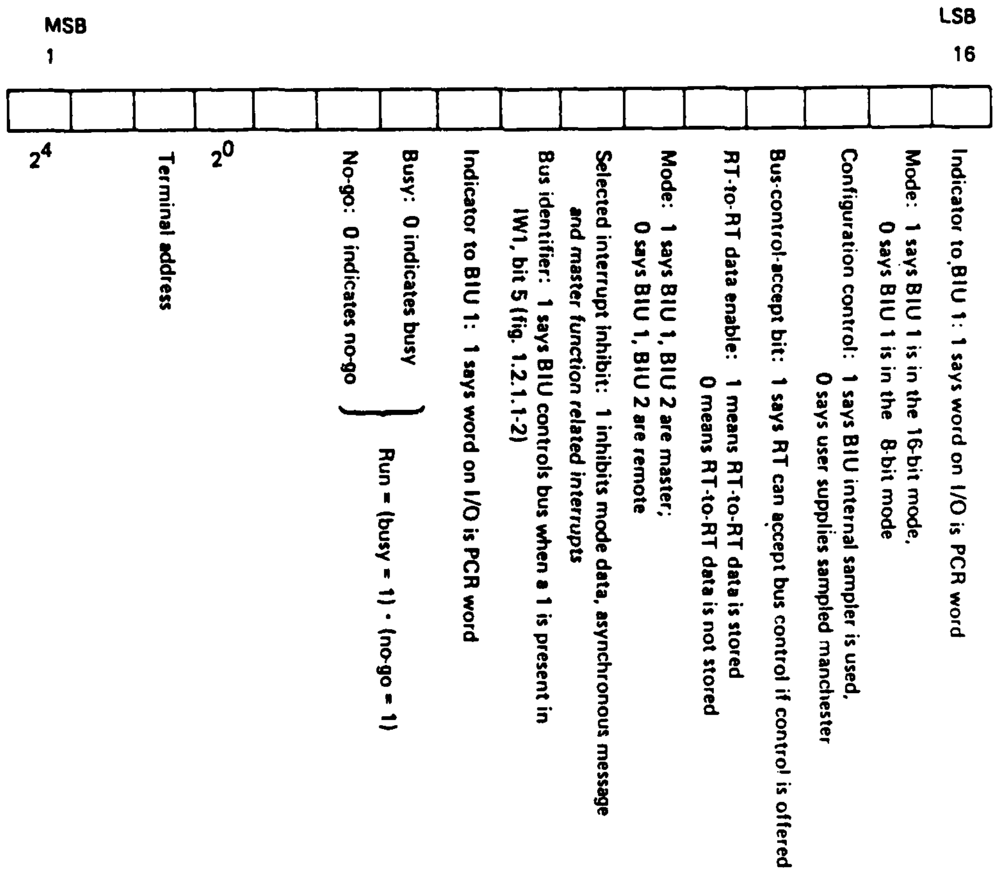


Figure 5.6-1. Processor Control Word Format

word (time tag, word count, and data validity). The pointer address value is incremented and the value loaded into the BIU's address register for use when it executes its DMA transfers.

Once the data buffer address is set up, it is ready to transmit the command word. From that point, the BIU handles data transfer via its interface. If the message is an RT receive message, the data transfers by the BIU complete the message process; however, if the message is an RT transmit message that is received by the BIU, the data transfers to memory are followed by a final DMA transfer of the tag word into the first address of the data buffer. This process is summarized in table 5.6-2. The RT operation is summarized in table 5.6-3.

5.6.2 BIU Control Instruction Interface

Control instructions are used by the processor to initialize and to operate on the BIU. Control instructions occur either via programmed I/O or via memory-mapped I/O. The control instructions in this example are treated as memory-mapped I/O instructions. These registers in the BIU are read or written much the same as the processor reads or writes its memory.

Table 5.6-2. Summary of Bus Controller BIU Message Processing

- BIU DMA's instruction words 1, 2 from host.

- BIU generates command word.

- BIU appends T/R and subaddress bits of command word to the least significant end of a 10-bit base address to form a 16-bit address into a pointer table:

		LSB
BASE ADDRESS (10 BITS)	T/R	SUBADDRESS (5 BITS)

- BIU DMA's pointer-from-pointer table. The pointer is the location of the first address in the data buffer.

- Pointer is stored in the BIU's pointer register.

- BIU loads incremented value of pointer into external address register.

- BIU transfers command word.

- BIU handles data DMA's.

- In the case of a RT transmit message, the final data DMA by the BIU is followed by a DMA of the tag word into the first address of the buffer. The transferred tag word contains the minor cycle number, word count, and the data error bit:

MINOR CYCLE NUMBER (10 BITS)	WORD COUNT (5 BITS)	DATA ERROR

Table 5.6-3. Summary of Remote Terminal BIU Message Processing

- BIU receives an RT transmit or an RT receive command.
- BIU determines that a command word is present.
- BIU determines if the command is an RT transmit or an RT receive command and begins data buffer address generation.
- BIU appends the T/R and subaddress bits of the command word to the least significant end of a 10-bit base address to form a 16-bit address into a pointer table.
- BIU DMA's pointer-from-pointer table. The pointer is the location of the first address in the data buffer.
- Pointer is stored in the BIU's pointer register.
- BIU loads incremented value of pointer into external address register.
- BIU handles data DMA's.
- In the case of an RT receive message, the final data DMA is followed by a DMA of the tag word into the first address of the data buffer.

The control codes available to the host are:

0000	Load	Mode data register	(MDR)	
0001	Load	Master function register (master)	(MFR)	
0010	Load	Instruction address register	(IAR)	
0011	Load	Base address register	(BAR)	
0100	Load	Processor control register	(PCR)	(both BIUs)
0101	Load	Status word data	(SWD)	
0110	Load	Built-in-test register	(BIT)	
0111	Halt	(graceful)		
1000	Output	MDR		
1001	Output	Receive status (master)/MFR (RT)		
1010	Output	IAR		
1011	Output	Transmit status (master)/BAR (RT)		
1100	Output	PCR		
1101	Output	ISR		
1110	Output	BIT		
1111	Abort			

The instruction, load mode data register (MDR), allows an RT or controller a place for storage of some special control word to be used in a later transfer. The controller BIU's processor might, for example, load bus designator information in the BIU's MDR and then direct the BIU to issue a mode command to an RT requiring it to shut down a selected transmitter. The controller would send the command word, followed by data from its MDR, to the RT.

The RT BIU's processor might, for example, load a service designator (or vector word) in its MDR using the load MDR instruction. It would then set a service request (SR) flag using the load status word instruction. Detecting the flag set in the status word, the controller BIU would generate an interrupt. The interrupted host, determining the interrupt cause, would request the vector word. When the transmit vector word mode command was received by the RT, the BIU would use the detection of the command as a reset signal for the SR and subsystem fail (SF) flag and return the 14-bit

vector and SR and SF bits contained in the MDR. The resetting of the SR and SF flag has no effect on the contents of the MDR, which eliminates message errors affecting the eventual acquisition of the vector word through automatic retries.

The instruction, load master function register (MFR), allows a controller BIUs host processor to update the timing information (e.g., minor cycle number) used by the BIU when the BIU time-tags buffer data. The MFR data can also be transferred from the controller BIU's MFR to an RT's MFR by a mode command (synchronize). If this mode command is broadcasted, all the RT BIUs are updated simultaneously.

The instructions affecting the instruction address register and the base address register, respectively, load a pointer to the set of BIU instructions and load the first 10 bits of the address of the message area.

The processor loads the BIU PCR to indicate to the BIU the BIU ID, bus ID (bus A or bus B controller), mode of operation (RT or controller), ability to accept bus control bus status, GO/NO GO status, and busy status.

The instruction, load status word data, allows for the setting or resetting of the subsystem flag or the service request. These bits, at the disposal of the RT processor, allow for additional (asynchronous) service from the controller beyond the periodic commands.

The instruction, load built-in-test register, allows the RT to report non-message-related failures (e.g., DMA handshake error) to the controller. Typically the controller, as part of a recovery procedure, would read the RT's BIT register (see fig. 5.6-2).

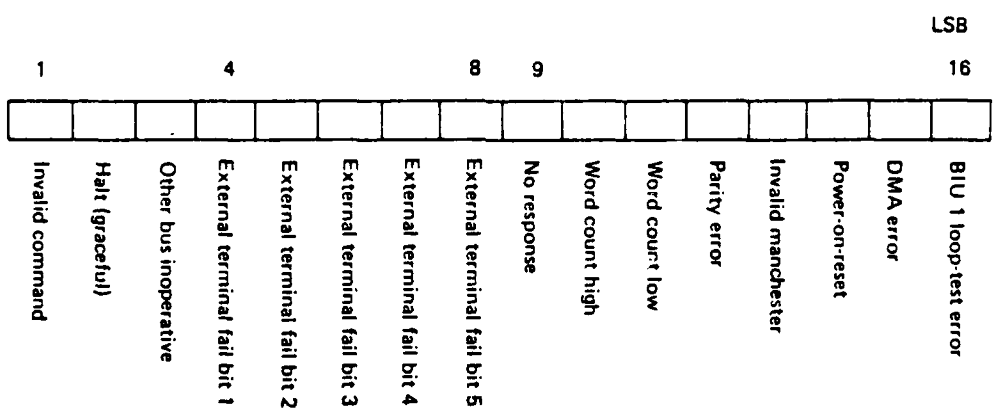


Figure 5.6-2 Bit Word Format

5.6.3 Interrupt Interface

As mentioned above, the BIU sets interrupts on various conditions:

a. Message errors
b. Status word exceptions
c. Certain mode commands
d. Program requirements

Interrupt generation reflects one of several facts:

a. The BIU has encountered a Manchester bus data transfer problem and error indications cannot be overcome without host intervention.

b. The BIU has been initialized because of a power dropout or startup and needs to be set up by the host.

c. The BIU has finished the bus-oriented tasks required of it by the BIU program in host memory and the program required host notification.

d. The host decides to intervene in BIU operation and commands the BIU to halt the operation gracefully.

As the word itself indicates, an interrupt is a break in an ongoing operational scenario, and when such a break occurs some trace of what happened must be recorded. Assume that each BIU possesses its own interrupts, so that the BIU that is interrupted can be identified. The possible reasons for interrupt generation are recorded by the BIU in the BIU's ISR (see fig. 5.6-3) and BIT (see fig. 5.6-2) registers. Both these registers are available to the BIU host. These interrupts can be viewed from the perspective of the master controller or of the remote terminal. Both aspects are covered below.

5.6.3.1 Interrupts Generated in the Master Controller

Message errors detected by the BIU include:

a. Manchester biphase errors
b. Word parity errors
c. No response
d. Message too short
e. Message too long

The BIU may diagnose a message error caused by failure of the above criteria. For example, an error could have occurred so that the sync detection circuitry failed to validate the last data word of a given message because of distortion. Because the BIU did not detect the last word, it registers an error in message since the message was too short. The BIU has automatic retry capability and can be programmed for up to three additional message communication attempts, so hard failures tend to be separated from random error occurrences suggested by this example. In the case of a hard failure, the BIU will exhaust the retry attempt(s) and still find that a message error is present. When the BIU is involved in a message sequence, the BIU saves any indication of error-present until the message is completed. At the end of the message execution, the error present flag

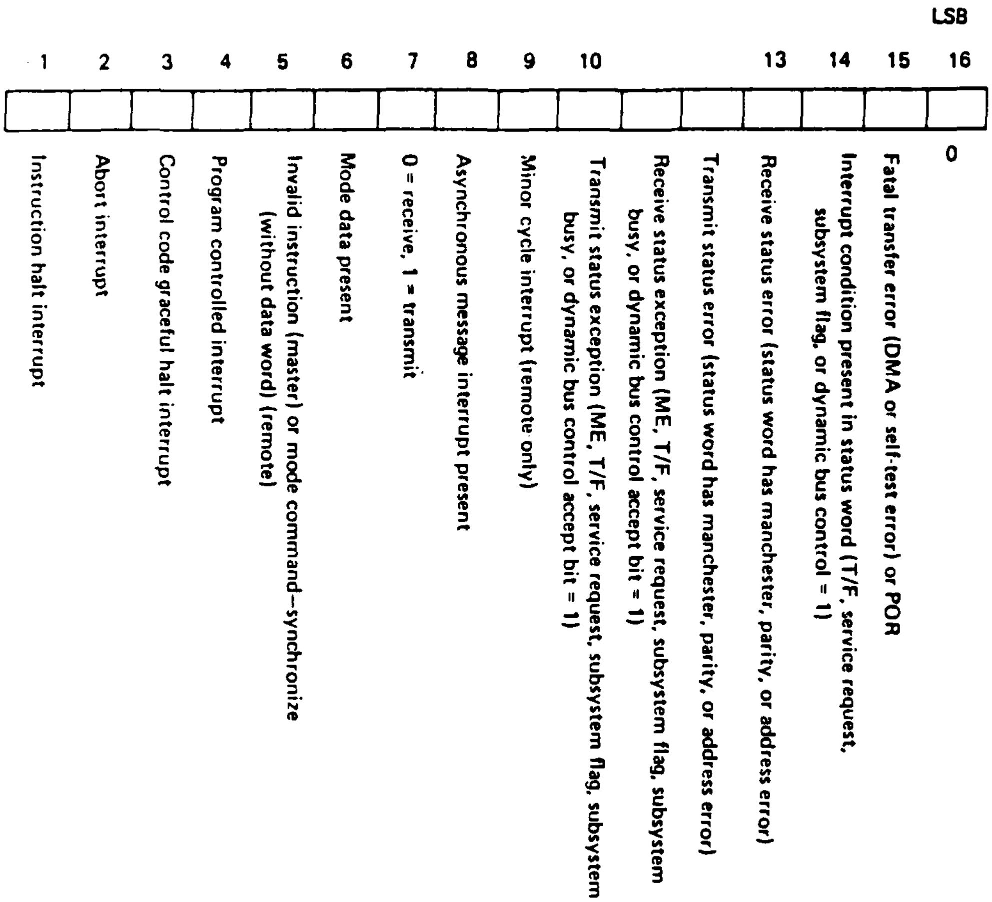


Figure 5.6-3. Internal Status Register

prompts the BIU to transfer the error word data (representing specific message errors, power-on reset, DMA error, and the loop-test error) into bits 4 through 16 of its own BIT register. After the transfer, the BIU tests for the presence of power-on reset, DMA errors, or loop-test errors. If power-on reset, DMA error, or loop-test error has occurred or the retry count is zero and a message error occurred, the BIU will interrupt the host and stop automatic operation. If none of these are present and the word count is not zero, the BIU clears the BIT word and ISR word and executes the next message sequence.

If the status word is valid but contains some exception (e.g., T/F, service request), the error present flag will interrupt the host without any retry attempt; however, two status word exceptions (ME and subsystem busy) can cause retry attempts. These status word exceptions indicate the receive command was not received by the terminal and that a retry could rectify the problem if allowed. So, like the error present flag, the two exceptional conditions found in the valid status word prompt the BIU to test the retry count and either execute another message sequence or generate an interrupt. Besides generating interrupts when message error or busy conditions prevent

successful communications, the BIU, acting as a master controller, generates interrupts in response to other conditions described below. In these cases, the BIU always stops operation until the interrupt is serviced. Under four conditions, the BIU will generate an interrupt when programmed to do so. The first condition can occur when the instruction pair being executed by the controller detects a message error or status word exception. Under these conditions, programmed message retries are attempted before generating the programmed interrupt. If the BIU is ultimately unsuccessful in accomplishing the instructed bus operation, the appropriate message-related conditions are recorded in bits 4 through 13 of the controller's BIT word, bit 4 of the ISR is set, and an interrupt is generated. It may be that during bus operation the BIU detects a loop-test error, DMA error, or power-on reset. In any of these cases, no retry is attempted, but the appropriate condition is recorded in bits 14 through 16 of the BIT word. Then, bit 4 of the ISR is set and an interrupt is generated. A second way of programming an interrupt is with the OP code of the instruction pair (see table 5.6-1). This can be set to require the BIU to halt. In this situation, the BIU decodes the requirement and executes it by setting bit 1 of the ISR and then generating the interrupt. A third method used to interrupt the host occurs when the BIU is given the control code 1111 by the host. This causes the BIU to stop all operations without regard to where it may be in its operating sequence. The BIU responds by setting bit 2 of the ISR and generating an interrupt. A fourth interrupt method is the graceful halt, which causes an interrupt based on the control code 0111 set by the host. The host can request the BIU halt operation, but in that case the BIU finishes its present operation, sets bit 3 in the ISR word, and interrupts the host. The graceful halt is executed identically to the program-controlled interrupt: only the indicator (in ISR bit 3 rather than the ISR bit 4) is different.

Interrupts are initiated by the BIU when the BIU discovers that an instruction pair contains the same device address in both instruction words. This check is made during command generation and, if this condition exists, the BIU operations are halted without ever beginning data bus transmission, and bit 5 of the ISR is set. Bit 6 of the ISR is set and an interrupt is generated by the BIU when a mode command requiring mode data has been executed. If, while attempting to acquire the mode data, the controller detects a message error or status word exception, programmed message retries are attempted if allowed. If the bus operation is ultimately unsuccessful, the appropriate message-related conditions are recorded in bits 4 through 13 of the controller's BIT word, bit 4 of the ISR is set, and an interrupt is generated.

Bits 7 and 8 of the ISR are associated with the execution of an asychronous message. Bit 8 indicates the BIU participated in an asychronous message and bit 7 indicates the BIU was the transmitter (bit 7 = 1) or the receiver (bit 7 = 0) of the message. The BIU processes these bits whenever it generates a command word with subaddress equal to 30. After executing the message associated with this command, the BIU generates an interrupt to the host. Treatment accorded message errors, specific status word exceptions, etc., is identical to that used for ISR bit 4 in the paragraphs above.

Bit 14 of the ISR is set and an interrupt is generated when the status word contains an interrupt condition. Treatment accorded message errors, specific status word exceptions, etc., is identical to that used for ISR bit

4 above. The conditions in the returned status word that interrupt the controller include T/R, service request, subsystem flag, and dynamic bus control acceptance. It is assumed that automatic message retries would not be scheduled in sensitive cases (e.g., use of the dynamic bus control mode command).

5.6.3.2 Interrupts Generated in the Remote Terminal

Section 5.6.3.1 describes how message errors detected by the BIU are transferred into the BIU's BIT register. This same process occurs in the remote terminal mode. Once a transfer is made, the remote terminal tests for the presence of power-on reset, DMA errors, or loop-test errors. If any of these are present, the BIU will generate an interrupt. These are the only message-error-related failures that can cause interrupt generation in the remote terminal.

Besides the above, the BIU as part of the remote terminal configuration generates interrupts in response to other conditions described below.

Bit 2 of the ISR is set and an interrupt is generated when the abort command (control instruction 1111) is given to the BIU. The host can use this command to stop all operation without regard to where the BIU may be in its operation (transmitting or receiving).

Bit 5 of the ISR is set and an interrupt is generated when the RT BIU receives a valid message containing the synchronize mode command (without data word).

Bit 6 of the ISR is set and an interrupt is generated when the RT BIU receives any of the mode commands, 10000 through 11111 (except 10001), provided that the T/R bit of the mode command is a zero. Under such conditions mode data are waiting for the host in the BIU's mode data register.

Bits 7 and 8 of the ISR are associated with the execution of an asynchronous message. Bit 8 indicates the BIU participated in an asychronous message. Bit 7 indicates the BIU was the transmitter (bit 7 = 1) or the receiver (bit 7 = 0) of the message. The BIU processes these bits whenever it receives a command word with subaddress equal to 30. After successfully executing the message associated with this command, the BIU generates an interrupt to the host.

Bit 9 of the ISR is set and an interrupt is generated when the synchronize mode command (with data word) is validly received. Upon reception of this command minor cycle time information is waiting for the host in the BIU's master function register.

CHAPTER 6

1553 EXAMPLES

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

List of Tables

6.0 MULTIPLEX SYSTEM EXAMPLES

At the present time many systems exist that use 1553 or a very similar data bus technique to implement subsystem communication. This section briefly describes some of these systems as examples of present 1553 implementation. The purpose of this section is to present these current (1980) examples of 1553 multiplexing in aircraft that will aid system engineers and hardware and software designers of future systems. Therefore, the point is to learn by example. Although the treatments of the systems included were written to a standard outline, there are differences because of the differences in the systems themselves and the information available.

The systems described were chosen as a representative cross section of existing systems. They range from simple to complex and include development, production, and laboratory systems. Implementations that closely follow the requirements of 1553(USAF) and 1553A are included, and all three branches of the military service are represented.

In addition to the above criteria, selection of examples was also influenced by availability of design details from either the contractor or the government sponsor. It was sometimes necessary to exclude a system solely because of insufficient data.

Another major consideration was the degree to which a system used 1553 compatible design. Many systems, though representing extensive use of multiplexing techniques, were eliminated because they were not closely related to 1553.

Exclusion of a system from this section is in no way a reflection on the quality of design or appropriateness for the intended application; it is simply a decision based on the selection criteria guidelines outlined previously.

6.1 F-16 MULTIPLEX SYSTEM -EXAMPLE 1

The F-16 is an air combat fighter supplied to the U.S. Air Force by General Dynamics, Forth Worth Division. The F-16 development program coincided closely with the initial publication of 1553(USAF) and so became the first vehicle to use and flight-test a 1553(USAF) compatible multiplex data bus system.

The F-16 data bus system is characterized by an extremely simple approach to architecture, bus control, redundancy management, mechanization, and bus control transition technique.

6.1.1 Application Area

The F-16 data bus is basically limited to the avionic system (AMUX) with essentially all major avionic subsystems using the bus for data transfer. In fact, the only major subsystem absent is the flight control system.

Nine different avionic subsystems interface directly to the F-16 data bus:

a. Fire control computer (FCC)
b. Fire control and navigation panel (FCNP)

c. Inertial navigation unit (INU)
d. Fire control radar (FCR)
e. Radar electro-optical display (REO)
f. Central air data computer (CADC)
g. Head-up display (HUD)
h. Stores management set (SMS)
i. Target identification set, laser (TISL)

All electronics required to interface with the multiplex bus are contained within the respective subsystem, thus completely eliminating the need for stand alone remote terminals (RT) or external signal conditioners. Thus, each subsystem provides data in digital form to the bus interface internal to the system, the only external signal interface being the serial multiplex bus.

6.1.2 System Architecture

Physical architecture of the F-16 avionic data bus system is shown in figure 6.1-1. A dual redundant bus network is used. The dual redundant bus is used primarily to prevent a single bus fau . (cable or connector) from rendering the system inoperative. With the exception of the SMS, none of the subsystems has functional redundancy. The SMS has two identical computers housed in one line replaceable unit (LRU). Each computer is connected to one of the 1553 buses through its own 1553 interface.

6.1.2.1 MIL-STD-1553 Compatibility

The F-16 data bus system was designed to be compatible with the interface requirements of 1553(USAF). Because 1553 does not contain sufficient information to specify procurable hardware, General Dynamics chose to include all of its multiplex data bus requirements in the F-16 interface control document (ICD).

In addition to the specification sheets normally included in an ICD, the F-16 ICD specification (16PP188) includes the essential requirements of 1553(USAF) plus additional details necessary to allow it to be used as a procurement specification. Supplementary requirements contained in 16PP188 include (1) definitions of the bits in the status word, (2) a description of the bus redundancy operation, (3) assignment of terminal addresses and subaddresses, and (4) necessary constraints on time coherence.

6.1.2.2 Deviations from MIL-STD-1553

The F-16 data bus system contains few actual deviations from 1553(USAF). Most would be more accurately classified as clarifications or additions to the original (no revision) standard. For example, the output voltage requirement was shifted, per 1553(USAF), from the bus side to the user (subsystem) side (16PP188) of the isolation transformer-resistor network to eliminate any supplier responsibility for the multiplex cable assembly itself, which was supplied by General Dynamics.

Other additions include the special use of all 1's in the address or subaddress fields. An all 1's terminal address is defined as a broadcast command and requires all subsystems to receive the data following the broadcast command without responding with a status word. Instead, upon

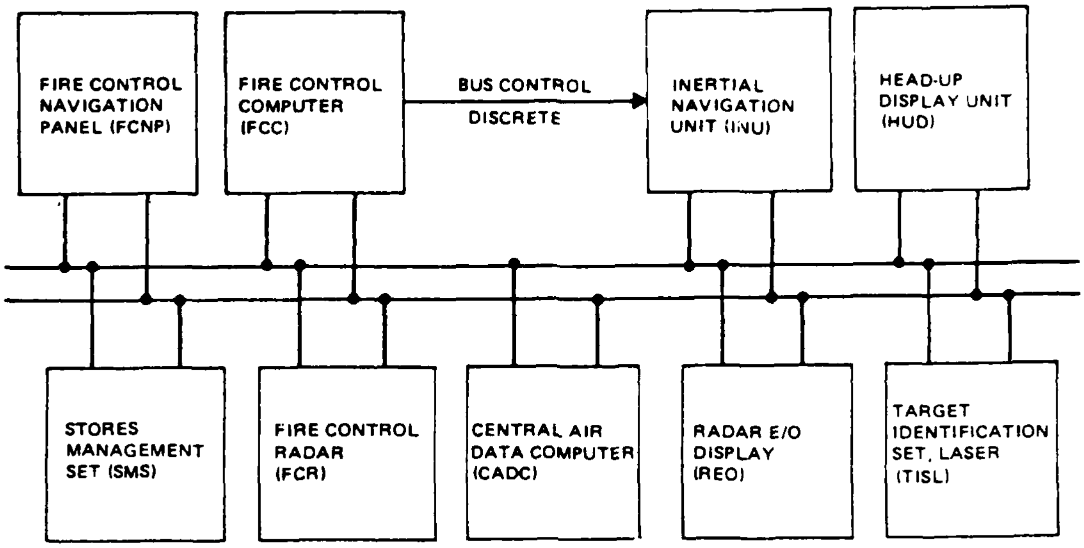


Figure 6.1-1. F-16 Multiplex System Architecture

receipt of a broadcast command, a broadcast function bit is set in the status register for transmission after the next normal command. Broadcast is specified in the ICD and implemented in the hardware; however, it is not used in the aircraft. All 1's in the subaddress/mode field designate a dedicated-function command determined by the contents of the word count field. These special codes are deviations from some, but not all, versions of 1553.

6.1.2.3 Multiplex Cable Assembly

The F-16 data bus uses a very short cable assembly. Although 1553(USAF) allows up to 300 ft of cable, the F-16 main bus is only 30 ft long. All subsystems are attached to the bus by stubs which are connected to the main bus by transformer-resistor coupling networks. Except for provisions for the TISL system, the stubs vary in length from approximately 2.5 ft to 16.7 ft. The TISL provision includes a 22.5 ft stub to an externally mounted PAVEPENNY pod.

Because of the modular assembly techniques used by General Dynamics, the isolation networks are mounted in matrices of multiple-stub terminations. The isolation networks are shielded from each other within these matrices. A typical F-16 cable assembly (bus A) with cable lengths is shown in figure 6.1-2.

6.1.2.4 Bus Protocol

The functional architecture of the F-16 multiplex data bus system is shown in figure 6.1-3. All transactions are command/response with bus control centralized in the FCC. A backup bus control capability resides in the INU. Controller-to-terminal, terminal-to-controller, and terminal-to-terminal exchanges, as defined by 1553, are used. Terminal addresses are

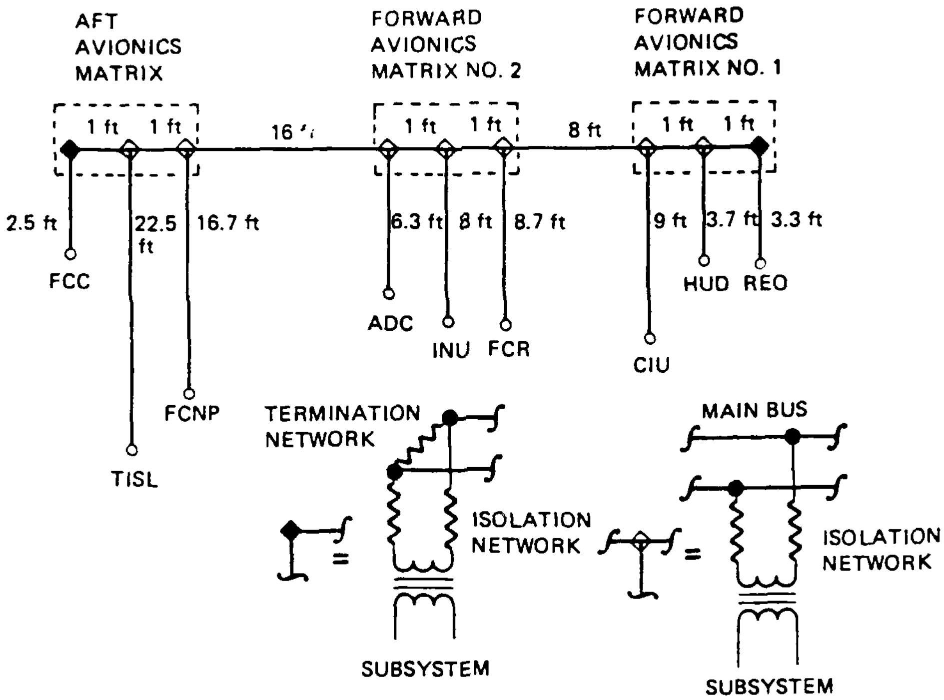


Figure 6.1-2. Typical F-16 Multiplex Bus Cable Assembly

hard wired within the remote terminals. Any subsystem is capable of receiving a command on either bus at any time. A subsystem always acts on the latest command word received. If a second command word is received (on either bus) while a previous message is being received, the subsystem interrupts receipt of the first message and accepts the latest command.This feature also allows a transmission on one bus to be interrupted by a subsequent command on the second bus.

All bus transactions are strictly scheduled by the FCC. Interrupt servicing as such is not allowed, but special servicing may be requested by a status word during a regular transaction.

Invalid Manchester word synchronization is used, with all subsystems using individual asynchronous clocks. A degree of synchronous operation is also possible as the central controller has the capability to command all terminal clocks to reset simultaneously. However this capability is not presently used. Instead, time-critical data, such as inertial platform measurements, is transmitted with a time tag so that individual users may establish latency of this data.

The use of time tag or algorithmic compensation has proved to be entirely satisfactory in removing the few variable latency problems encountered, thereby preserving the inherent simplicity of asynchronous operation.

Messages are transmitted in blocks ranging from 1 to 32 data words. The basic message transfer rate is 50 Hz with slower rates available in powers of two submultiples. The lowest data rate is 1.5625 Hz. Therefore, the major frame rate is 640 ms.

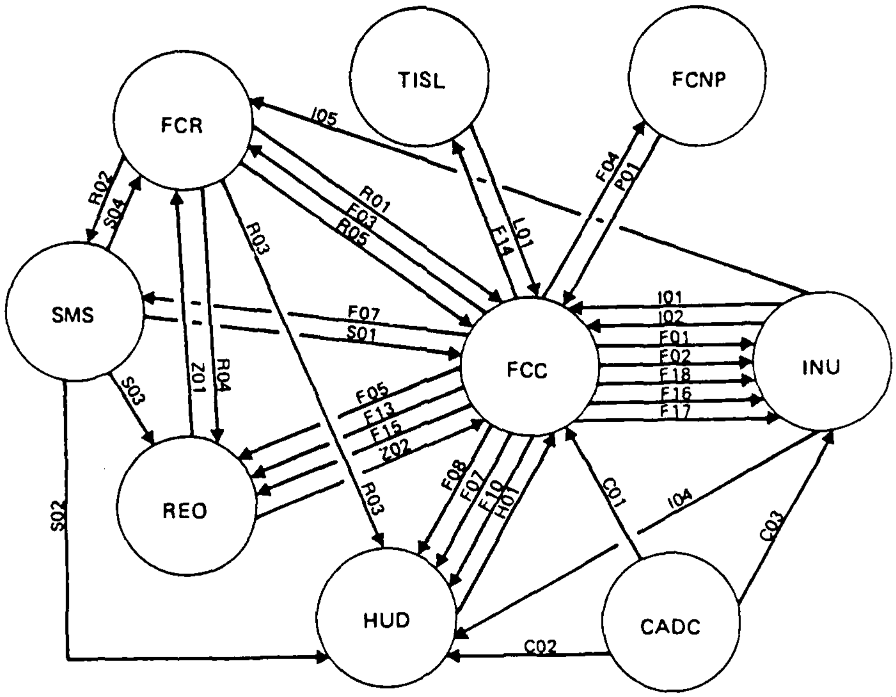


Figure 6.1-3. F-16 Multiplex Data Bus Functional Block Diagram (Primary Mode)

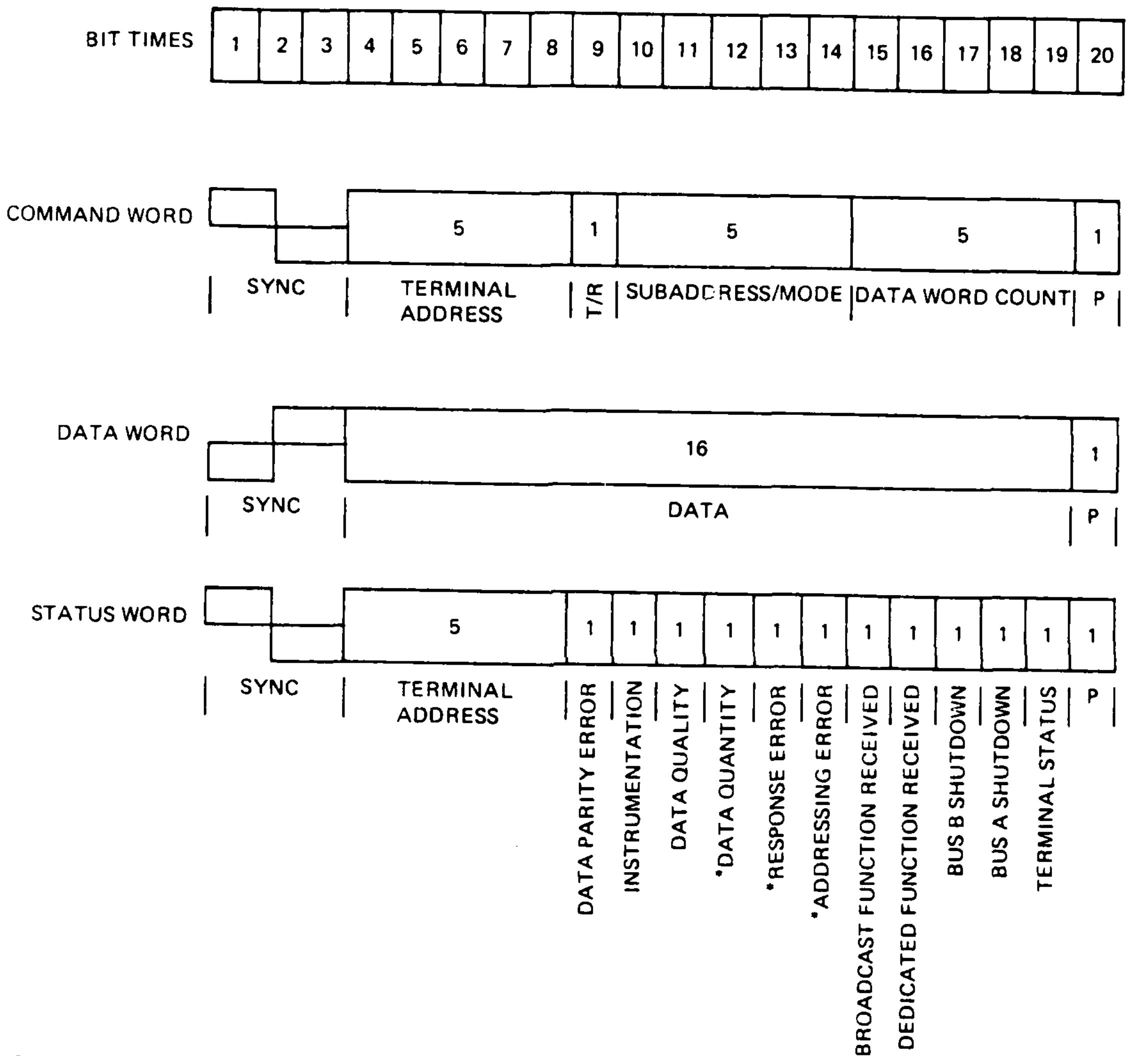
Command, status, and data word formats are as shown in figure 6.1-4. The command word is composed of a sync waveform, subsystem address, transmit/receive bit, subaddress/mode field, data word count, and a parity bit. Status word bit assignments are as shown. The data quantity, response error and addressing error bits (designated by * in fig. 6.1-4) are always transmitted as 0's on the bus. The bits may be set in the status word by the bus controller (FCC) after the word is received as an internal record of detected message completion failures in RT-to-controller transfer.

The subaddress/mode field in the command word is used to indicate subaddress or function commands per 1553. Subaddresses are used to identify specific data blocks to be transferred. Function commands are indicated when the subaddress/mode field contains all logic 1's.

Only two function commands (mode codes) are used by the F-16. These assignments are shown in figure 6.1-5. The transmit status command (00000) causes the subsystem to reset and initialize its receiver logic and respond with its status word only. In addition, if a subsystem receives a function command (mode code) with any bit pattern other than 00000 or 00001, the transmit status command is assumed.

6.1.3 System Control

The F-16 multiplex data bus system uses a simple control philosophy. The FCC, when operating, acts as the bus controller. If the FCC is not operating, the INU assumes bus control. This concept is further simplified by the restriction that the FCC can never operate as a remote terminal.



*Used only by FCC for internal status information.

Figure 6.1-4. Word Formats

Command interpretation	Word count field (Bit time)				
	15	16	17	18	19
Transmit status	0	0	0	0	0
Reset timer	0	0	0	0	1

Note: Any other bit pattern will cause RT to "transmit status."

Figure 6.1-5. F-16 Function Word Commands (Mode Codes)

6.1.3.1 Fault Isolation and Redundancy Management

All bus control is based on the ability to communicate. Communication status assessment is established through periodic polling of each terminal. Polling occurs at the basic frame rate of 1.5625 Hz. Based on the polling results data transfer with a subsystem is either established or deleted. If a subsystem responds to a poll, data communications are established. If the subsystem fails to respond for two consecutive polling periods, that subsystem's data transfer commands are deleted from the controller's current command table. Thus, periodic polling allows a subsystem's communication status changes to be discerned without reference to any additional input.

The ability-to-communicate approach also results in one of the simplest strategies possible for selection of the redundant channel. The controller simply always uses the channel that worked last. For example, a successful transfer on bus A would result in the next transfer of the same block also being attempted on bus A; however, if the first attempt on bus A failed, then the retry of that transmission would be attempted on bus B. Thus, communications will continue on the channel that is functioning. Note that the retry is limited to once per scan of the command table and is always initiated on the alternate bus. If the retry fails, the command is skipped. The sequence is "fail once, retry on alternate; fail twice, go to next command."

6.1.3.2 Primary and Backup Control

The F-16 FCC normally functions as bus controller. In case of an FCC power down or self-test failure, bus control is assumed by the INU. Once again, the simplicity of the F-16 AMUX system is apparent in the mechanization of the bus control transition technique. If the FCC is powered down or becomes inoperative, bus control is passed to the INU by a single discrete between the two units. During operation, the INU periodically samples the discrete for a high-voltage or high-impedance condition. If two consecutive samples are in the pass control or NO-GO state, the INU assumes bus control responsibilities. The INU then continues periodic sampling of the discrete and relinquishes control to the FCC immediately upon detection of a low-voltage GO condition on the discrete.

Bus control is greatly simplified in the backup mode by virtue because the FCC and TISL are completely eliminated from the system. This is apparent by comparison of the backup functional interface shown in figure 6.1-6 with that of the primary mode (fig. 6.1-3).

6.1.4 Bus Controller

The FCC's prominence in the primary data flow pattern (see fig. 6.1-3) figured heavily in the selection of the FCC as the primary bus controller. Also, General Dynamics was responsible for developing the operational software for the FCC, thus ensuring that the prime contractor maintained responsibility for system integration.

Again referring to figure 6.1-3 and following the previously established line of reasoning, it became apparent that the INU would figure most prominently in the data flow pattern should the FCC fail. Therefore, the INU was selected to perform the backup bus control function.

6.1.4.1 Primary Bus Control Implementation

Primary bus control resides in the FCC. The FCC is a Delco M362F computer procured for the F-16 and programmed by General Dynamics. Actual bus control is maintained by a microprogrammable hardware DMA controller that is initiated periodically by the FCC operational flight program (OFP). This controller, called the serial data interface (SDI) reads and executes bus transfer sequences stored in the FCC main memory. Once initiated by the FCC

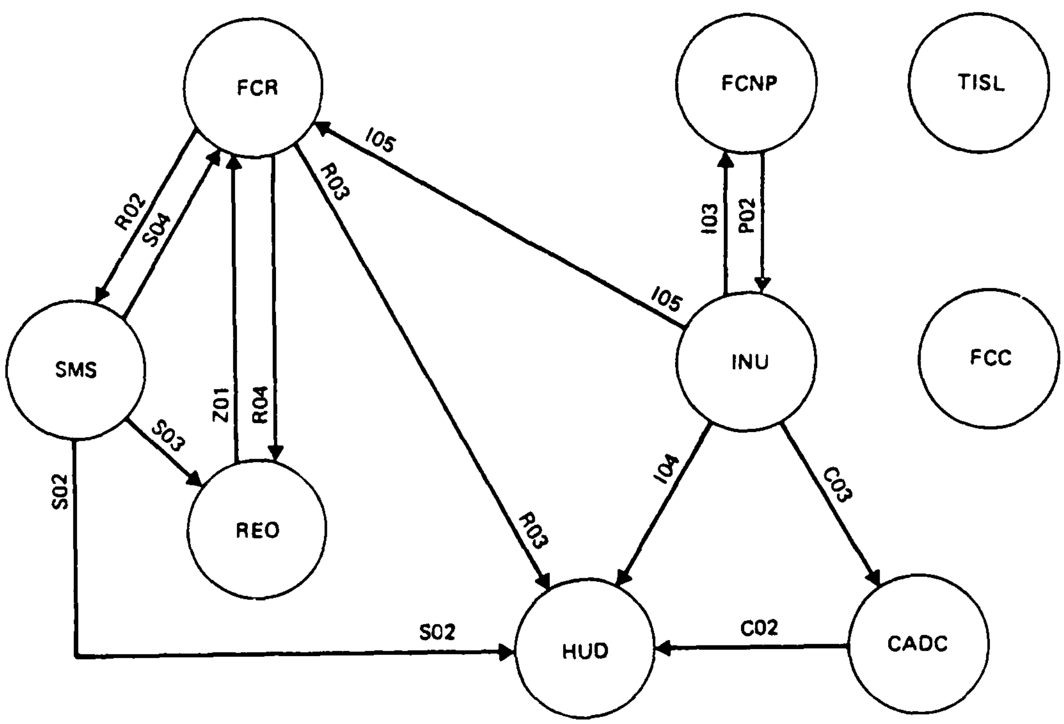


Figure 6.1-6. F-16 Multiplex Data Bus Functional Block Diagram (Backup Mode)

OFP software, the SDI will continue to read and execute the command table until either (1) the command sequence is complete or (2) a transmission fails to complete successfully. Software intervention is required only to start the SDI processor, analyze poll responses, and to process SDI-initiated interrupts such as "command failure" or "command task complete." A command failure interrupt is generated to the CPU when any commanded bus transmission fails to complete successfully. A command task complete interrupt is used twice per minor frame: (1) to inform the OFP that key input data has been received and (2) to indicate that all scheduled transfers for the current time frame have been completed. The first interrupt may be generated after any designated SDI command.

In addition to data transfers, the SDI may also be commanded (by the OFP software) to perform various internal functions such as command sequence branches, internal self-test, or SDI stop.

The OFP controls data transfers using a time-slice executive structure operating at a 50-Hz maximum computational rate. The SDI thus is commanded to initiate a transfer sequence once per 20 ms minor frame. Actual avionic interface data are structured into blocks with fixed transfer rates that are powers of two submultiples of the basic 50-Hz rate. Therefore, the command sequence in each minor frame will consist of data blocks from the 50-Hz group plus blocks from one of the submultiple groups. Because the lowest transfer rate is 1.5625 Hz, 640 milliseconds are required to complete a full data transfer sequence (major frame).

This periodic transfer sequence is implemented by linking the SDI command table prior to each minor frame. This is accomplished by setting two link commands to provide the desired path through the command table (fig. 6.1-7). The content of the command table is modified by the results of poll processing to eliminate the transfer commands of those subsystems that are not actively communicating.

Polling is accomplished at the major frame rate of 1.5625 Hz to ensure that satisfactory communication can be established with a terminal before a data transfer is attempted. Polling is accomplished using the F-16 dedicated function (mode code) command, which requests the addressed terminal to respond with its current status word only. The dedicated function command is distinguished by all "1's" in the subaddress/mode field.

Any poll command that fails (whether because of a reported fault status or a transmission failure) causes a CPU interrupt and subsequent software recording of the poll command failure. The result of each subsystem poll is then masked with the results of the previous poll and used to delete the data transfer for that subsystem from the command table if two successive poll failures are recorded. The next successful poll response from that subsystem will immediately reinstate the transfer command and so reestablish communication with the polled subsystem.

In addition to poll response errors, data transmission errors also are handled by special error-handling interrupt software. The error-handling software indexes an error response table that determines the appropriate error response for each command. The error response table will indicate (1) whether the command is to be retried, (2) the bus to be used for the retry, and (3) whether the transmitted data (if any) should be invalidated. As

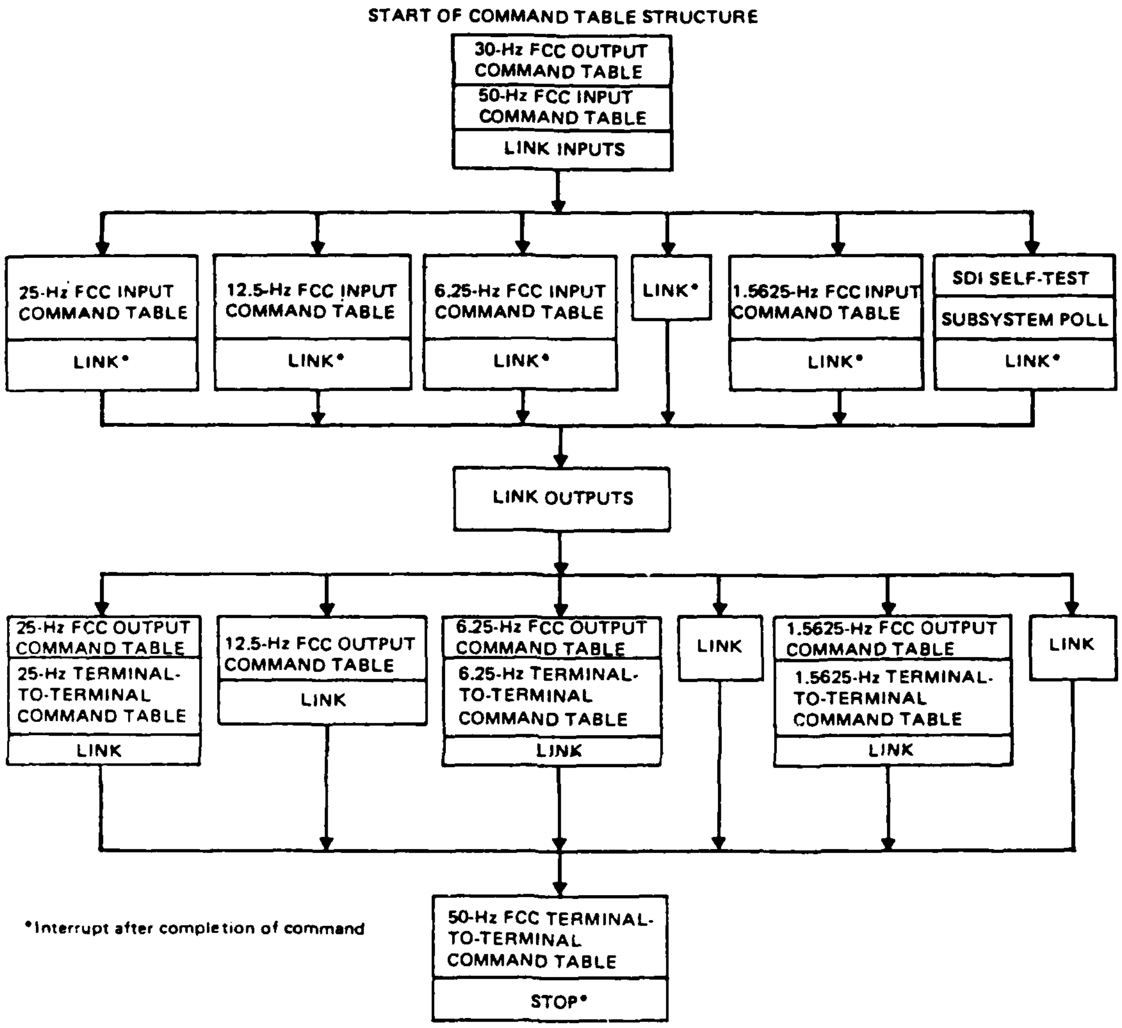


Figure 6.1-7. Bus Control Command Table Structure

previously noted, all retries are currently initiated on the alternate bus, but sufficient software flexibility exists to allow the retry mode to be selectively changed on an individual command basis if so desired. The FCC used 950 16-bit words of memory for bus control. This includes the complete control algorithm for (1) SDI start commands, (2) interrupt handler for command link selection, (3) error interrupt handler, (4) poll analysis module, and (5) bus command tables. CPU processing time comprises less than 2% of the machine duty cycle.

6.1.4.2 Secondary Bus Control Implementation

Secondary or backup bus control is provided by the INU. The INU periodically samples the bus control discrete from the FCC and assumes bus control after two consecutive NO-GO samples. The basic bus control concept

used by the INU is essentially the same as that previously discussed for the primary controller. The backup control algorithm, however, is considerably simpler than that of the primary system for two reasons. First, the number of data blocks to be managed is much smaller. Second, fault reporting and error recovery requirements are considerably reduced. The INU contains a hardware DMA controller that is commanded by the INU OFP software in the same manner as the FCC. The only difference in implementation is that backup redundancy management is hardware controlled by the INU, whereas OFP software in the FCC controls redundancy management in the primary bus control mode.

6.1.5 Remote Terminal

The F-16 multiplex data bus system interfaces with and provides complete communication with nine major subsystems, as listed in section 6.1.1. All bus interfaces are integral to the subsystems they serve. This approach drastically reduces integration problems associated with standalone RT/signal conditioning systems.

Of the nine subsystems, seven always act as RT's only. One, the FCC, acts as the primary bus controller and is deleted from system communication in the backup mode. It can never act as a RT. The INU can act as either a RT (in the primary mode) or a bus controller (in the secondary mode).

6.1.5.1 Subsystem Interface

Because all bus interfaces are integral to the subsystems that they serve, the usual subsystem interface is solely the responsibility of the avionic's supplier and, in fact, does not exist external to the subsystem. Thus, the communication interface with the subsystem is limited to the 1553 bus. None of the F-16 subsystems use a standard interface module. The bus interfaces within the various subsystems represent independent designs by six different suppliers.

6.1.5.2 Fault Isolation and Redundancy

Although the F-16 has a dual redundant bus network, only the stores management set is fully dual redundant. None of the other subsystems have functional redundancy. The SMS utilizes two identical AMUX interface modules. Each AMUX module serves one redundant half of the SMS and communicates with one of the two data buses.

Fault conditions within a subsystem which could affect data validity are "OR'ed" into a single terminal status bit which is returned in the status word. Fault conditions included in the terminal status bit are determined on an individual subsystem basis. The basic ground rule, however, is that to be included in the terminal status bit, the failure must affect validity of all data transmitted by the subsystem. Other fault conditions are detected by the system controller based on ability to communicate.

A 1553 feature that is implemented in each F-16 terminal is "operation on latest command word." This feature requires that a terminal act on the latest command word received on either bus even though it may interrupt receipt or transmission of a message in progress. A message being received may be interrupted on either bus. A message being transmitted may be

interrupted by a command on the alternate bus. This is the only condition in the F-16 system under which both buses may be in use simultaneously. This feature is potentially useful as a priority override or to shut down a faulty transmitter.

A terminal status word only may be requested from an individual terminal by use of the dedicated function command designated by an all 1's subaddress/mode code field, as described in section 6.1.2.4.

6.2 LAMPS MULTIPLEX SYSTEM -EXAMPLE 2

The light airborne multipurpose system (LAMPS) is a Navy ship and helicopter system that uses the helicopter as an airborne platform to extend the range of the destroyer's sensors and weapons. The shipboard and airborne elements communicate over a two-way digital data link. The airborne avionic system is interconnected by two 1553A data bus channels.

6.2.1 Application Area

The LAMPS airborne system consists of seven major subsystems:

a. Antisubmarine warfare
b. Weapons and armament delivery
c. Communications
d. Data handling
e. Antiship surveillance and targeting (ASST)
f. Navigation
g. Data display

System integration and control is provided by an AN/AYK-14 standard airborne computer (SAC). The SAC performs system integration and control functions for the avionic system. Four I/O channels are used for communication between the AN/AYK-14 computer and the avionic subsystems. Two of these channels are implemented by 1553A data buses. The I/O channels are designated as follows:

a. Avionic system 1553A channel (logical channel 0)
b. Program load 1553A channel (logical channel 1)
c. Discrete channel (logical channel 15)
d. PROTEUS digital channel (logical channel 2)

Only two major LAMPS subsystems are not connected to the data bus: the magnetic anomaly detector (MAD) and the radar.

6.2.2 System Architecture

The physical architecture of the LAMPS airborne system is shown in figure 6.2-1. Three 1553A data buses comprising two channels are used. Channel 0, the avionic system channel, is referred to as the LAMPS data bus. It connects standard airborne computer number one (SAC-1) with the 10 RT's that comprise the avionic system. The RT's are located in the following equipment:

a. Radio terminal set (data link), AN/ARQ-44
b. Electronic warfare support measures (ESM) set, AN/ALQ-142

used by the INU is essentially the same as that previously discussed for the primary controller. The backup control algorithm, however, is considerably simpler than that of the primary system for two reasons. First, the number of data blocks to be managed is much smaller. Second, fault reporting and error recovery requirements are considerably reduced. The INU contains a hardware DMA controller that is commanded by the INU OFP software in the same manner as the FCC. The only difference in implementation is that backup redundancy management is hardware controlled by the INU, whereas OFP software in the FCC controls redundancy management in the primary bus control mode.

6.1.5 Remote Terminal

The F-16 multiplex data bus system interfaces with and provides complete communication with nine major subsystems, as listed in section 6.1.1. All bus interfaces are integral to the subsystems they serve. This approach drastically reduces integration problems associated with standalone RT/signal conditioning systems.

Of the nine subsystems, seven always act as RT's only. One, the FCC, acts as the primary bus controller and is deleted from system communication in the backup mode. It can never act as a RT. The INU can act as either a RT (in the primary mode) or a bus controller (in the secondary mode).

6.1.5.1 Subsystem Interface

Because all bus interfaces are integral to the subsystems that they serve, the usual subsystem interface is solely the responsibility of the avionic's supplier and, in fact, does not exist external to the subsystem. Thus, the communication interface with the subsystem is limited to the 1553 bus. None of the F-16 subsystems use a standard interface module. The bus interfaces within the various subsystems represent independent designs by six different suppliers.

6.1.5.2 Fault Isolation and Redundancy

Although the F-16 has a dual redundant bus network, only the stores management set is fully dual redundant. None of the other subsystems have functional redundancy. The SMS utilizes two identical AMUX interface modules. Each AMUX module serves one redundant half of the SMS and communicates with one of the two data buses.

Fault conditions within a subsystem which could affect data validity are "OR'ed" into a single terminal status bit which is returned in the status word. Fault conditions included in the terminal status bit are determined on an individual subsystem basis. The basic ground rule, however, is that to be included in the terminal status bit, the failure must affect validity of all data transmitted by the subsystem. Other fault conditions are detected by the system controller based on ability to communicate.

A 1553 feature that is implemented in each F-16 terminal is "operation on latest command word." This feature requires that a terminal act on the latest command word received on either bus even though it may interrupt receipt or transmission of a message in progress. A message being received may be interrupted on either bus. A message being transmitted may be

interrupted by a command on the alternate bus. This is the only condition in the F-16 system under which both buses may be in use simultaneously. This feature is potentially useful as a priority override or to shut down a faulty transmitter.

A terminal status word only may be requested from an individual terminal by use of the dedicated function command designated by an all 1's subaddress/mode code field, as described in section 6.1.2.4.

6.2 LAMPS MULTIPLEX SYSTEM -EXAMPLE 2

The light airborne multipurpose system (LAMPS) is a Navy ship and helicopter system that uses the helicopter as an airborne platform to extend the range of the destroyer's sensors and weapons. The shipboard and airborne elements communicate over a two-way digital data link. The airborne avionic system is interconnected by two 1553A data bus channels.

6.2.1 Application Area

The LAMPS airborne system consists of seven major subsystems:

a. Antisubmarine warfare
b. Weapons and armament delivery
c. Communications
d. Data handling
e. Antiship surveillance and targeting (ASST)
f. Navigation
g. Data display

System integration and control is provided by an AN/AYK-14 standard airborne computer (SAC). The SAC performs system integration and control functions for the avionic system. Four I/O channels are used for communication between the AN/AYK-14 computer and the avionic subsystems. Two of these channels are implemented by 1553A data buses. The I/O channels are designated as follows:

a. Avionic system 1553A channel (logical channel 0)
b. Program load 1553A channel (logical channel 1)
c. Discrete channel (logical channel 15)
d. PROTEUS digital channel (logical channel 2)

Only two major LAMPS subsystems are not connected to the data bus: the magnetic anomaly detector (MAD) and the radar.

6.2.2 System Architecture

The physical architecture of the LAMPS airborne system is shown in figure 6.2-1. Three 1553A data buses comprising two channels are used. Channel 0, the avionic system channel, is referred to as the LAMPS data bus. It connects standard airborne computer number one (SAC-1) with the 10 RT's that comprise the avionic system. The RT's are located in the following equipment:

a. Radio terminal set (data link), AN/ARQ-44
b. Electronic warfare support measures (ESM) set, AN/ALQ-142

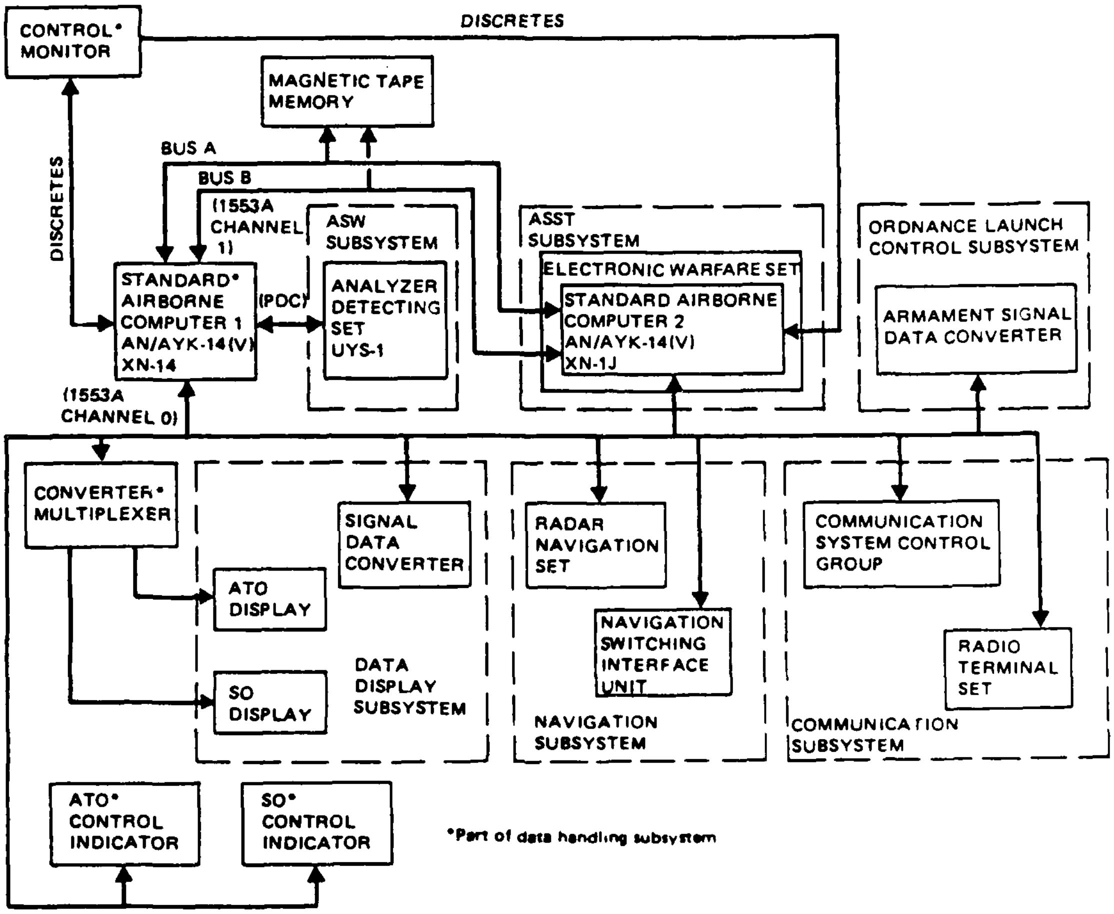


Figure 6.2-1. LAMPS MK-III Multiplex System Architecture

c. Signal data converter (radar), OU-103A
d. Communications system control group, OK-374/ASC
e. Radar navigation set (doppler), AN/APN-217
f. Converter-multiplexer (CMUX), CV-3435/A
g. Navigation switching interface unit (NSIU), SA-2213/ASQ
h. ATO control indicator (keyset), C-10487/ASQ-164
i. SO control indicator (keyset), C-10486/ASQ-164
j. Armament control indicator set (ACIS), AN/ASQ-165

A single (nonredundant) 1553A data bus connects these subsystems to SAC-1, which serves as the primary bus controller.

The other 1553A channel (channel 1) is used as program the load and data extraction path and connects the two AN/AYK-14 computers (one within the ASST subsystem) to the magnetic tape memory (MTM). This channel is implemented with dual redundant buses, designated bus A and bus B. Redundant bus selection is a function of the bus controller software.

6.2.2.1 MIL-STD-1553 Compatibility

The LAMPS data buses were designed to be compatible with 1553A. The prime contractor for avionic integration (IBM) issued a detailed specification, "LAMPS Data Bus Interface Requirements" (IBM 6226114C), which defines the requirements and characteristics of this interface. This document is intended to amplify and interpret 1553A. More detailed information concerning data word definition is contained in "Interface Characteristics Specification" (IBM 6226054).

6.2.2.2 Deviations from MIL-STD-1553

The only apparent deviation from 1553A is that the LAMPS MK-III system does not use coupler boxes to connect the bus to the stubs. While this is a deviation from 1553A, it complies with 1553B. The detailed specification (6226114C) specifically deletes paragraphs 4.2.4.5, 4.2.4.6 and 4.2.5.1 of 1553A and replaces them with an alternate coupling and termination configuration.

Since each LAMPS terminal is contained within a subsystem, section 4.4 of 1553A is also deleted.

Direct terminal-to-terminal transfers are not used by the LAMPS system. This type of transfer is allowed but not required by 1553A.

6.2.2.3 Multiplex Cable Assembly

The channel 1 data bus assembly is relatively short and is limited to only three terminations: SAC-1, SAC-2, and the MTM. Channel 0 however, has a total length of approximately 80 ft and has 11 terminations including the bus controller. Cost was the driving factor in limiting the channel 0 bus to a single cable, rather than the more common dual redundant pair.

To reduce weight and also reduce the reliability risk because of additional connectors, coupler boxes were not used to connect the stubs to the bus. Instead, a daisy-chain type of connection was used, as shown in figure 6.2-2. The stub length from the RT receptacle to the associated dc isolation and coupling transformer is limited to 10 in by specification 6226114C. The data bus in and out connections are spliced within the backshell of the data bus connector. A maximum of 2 in is allowed within the backshell to protect the splices. While this coupling method is a deviation from 1553A, it is allowed by 1553B. All data bus wiring is specified as twisted-shielded pair with a nominal impedance of 70 ohms.

6.2.2.4 Bus Protocol

All transactions on the LAMPS data bus are command/response with bus control centralized in the AN/AYK-14 computer (SAC-1). A backup bus control capability resides in SAC-2, a slightly different configuration of the AN/AYK-14 that is normally used as a RT in the ASST subsystem to process electronic warfare sensor signals. If SAC-1 fails, its avionic operational program (AOP) is loaded into SAC-2 from the MTM. In this configuration, electronic warfare signals can no longer be processed and SAC-2 serves as a backup bus controller in a degraded system mode.

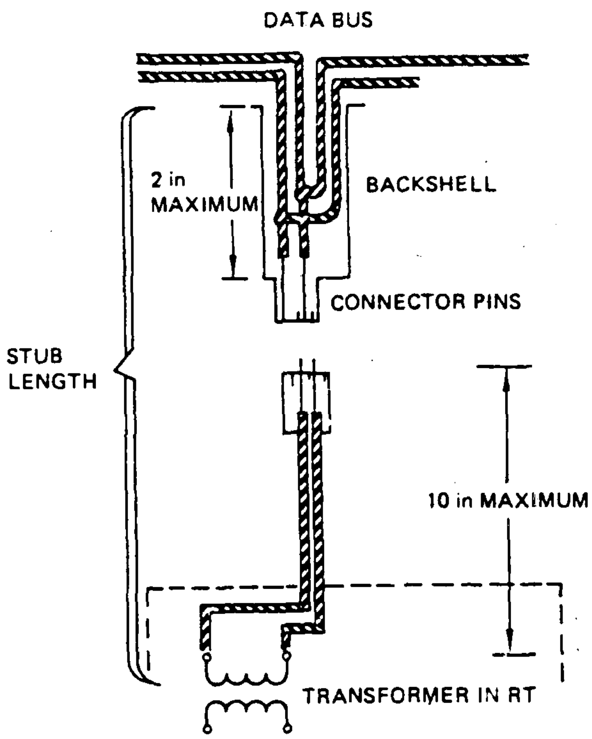


Figure 6.2-2. LAMPS MK-III Stub to Bus Coupling Method

Controller-to-terminal and terminal-to-controller exchanges, as defined by 1553, are used. Terminal-to-terminal transfers are not used by the LAMPS data bus (channel 0). The functional block diagram (fig. 6.2-3) summarizes the functions of the controller and RT.

SAC-1 and the RT's also communicate over a parallel discrete channel. This channel provides the bus controller eight external interrupts, 32 bidirectional signals (input or output, selectable by the controller program), and 32 input-only signals. The controller uses this channel to issue master clear signals to the RT's, and to accept RT power on and off status and the reflection of the master clear signal. The master clear signal is used for initialization and control of a faulty RT. This channel is also used for program load control and initialization procedure, and communication between the two SAC's and the control monitor, for program load control and initialization procedures.

The AOP in the AN/AYK-14 (SAC-1) includes an algorithm that controls the periodic communication between the computer and the various subsystems. Messages are transmitted in blocks ranging from 2 to 32 data words. The basic message transfer rate is 25 Hz with slower rates available in powers of two submultiples. The lowest message transfer rate is 1.5625 Hz. Figure 6.2-4 illustrates the LAMPS command word format and associated subaddress/mode field assignments. Only all 0's (00000) in the mode field

is reserved by 1553A. This value signifies that the word count/mode code field does not contain a word count but a mode code. The assignment of the 32 values of this mode code field is defined by specification 6226114C.

When the subaddress/mode field is set to 0 and the word count/mode code field is 2, the RT transmits an RT status word as specified by 1553. A unique feature of the LAMPS system is the definition of a special RT built-in-test (BIT) word. This transmission is specified by a subaddress/mode field value of 2 and causes the addressed terminal to respond with an RT BIT word. Specific faults in the bus message traffic and built-in-test (BIT) information are transferred to the controller by the RT BIT word. The structure of this word is shown in figure 6.2-5.

The RT BIT word has a 5-bit message error field that is defined as shown. When a bit is set in this field, the RT will also set the message error bit in the RT status word. The remaining 11 bits are associated with subsystem faults and are found in two fields. The information in these fields is uniquely defined for each RT. Fault indications in these subsystem fields will cause the terminal flag (T/F) bit in the status word to be set.

Because the RT BIT word is a data word, the 1553A scheme permits as many of these words as required. The number of BIT words varies from RT to RT and depends on the amount of fault information associated with the equipment connected to the RT. As currently designed, four is the greatest number of BIT words that the bus controller sees in response to a BIT mode command.

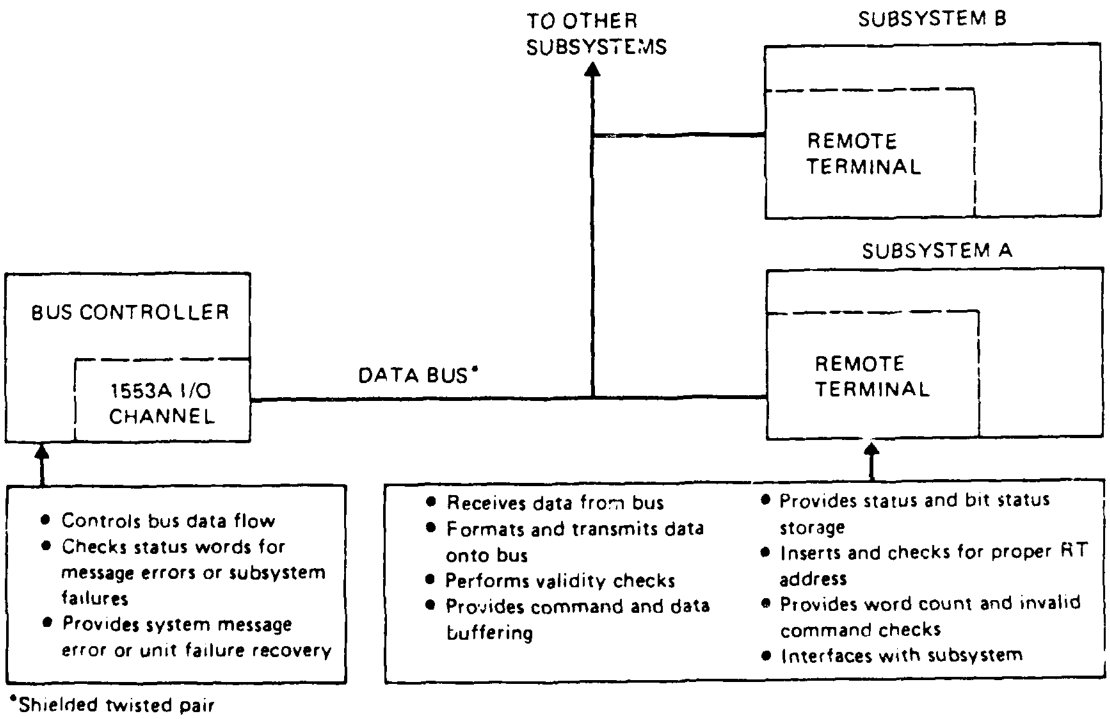
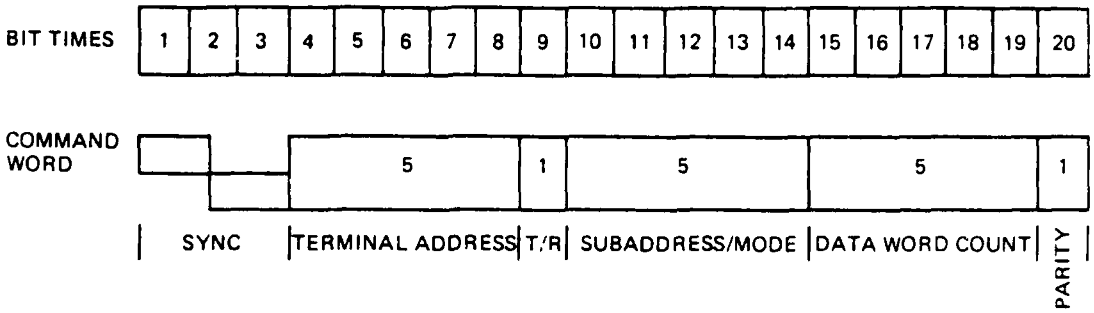


Figure 6.2-3. MIL-STD-1553A Data Bus Functional Block Diagram



Where:

T/R flag indicates the direction of the data flow
= 0 controller to the remote terminal
= 1 remote terminal to the controller

Subaddress/mode (SAM) field is defined as:

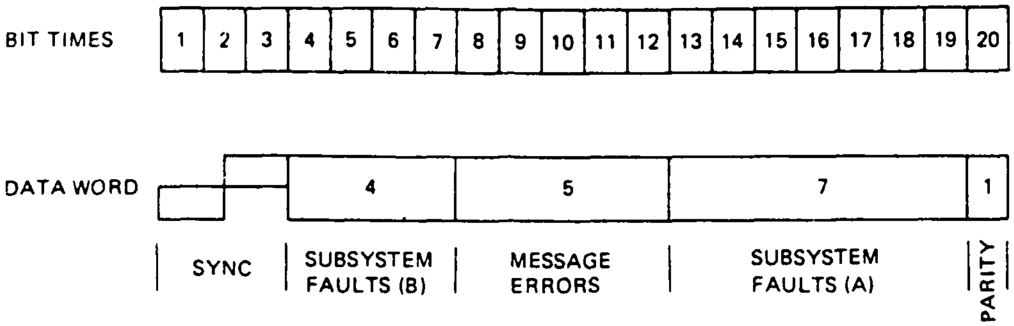
Value	Bit pattern (bits 10-14)	Definition
0	0 0 0 0 0	Mode/discrete data
1	0 0 0 0 1	Normal data transfer
2	0 0 0 1 0	Transmit bit status word(s)
3	0 0 0 1 1	Retransmit last message
4	0 0 1 0 0	Transmit last command
5	0 0 1 0 1	Control command data transfer
6	0 0 1 1 0	Multimessage transfer

Data word count field is the word count of the number of data words to be transferred, except for use where the subaddress/mode field is zero. For this case, only the following definitions will be used.

Value	Bit pattern	Definition
0	0 0 0 0 0	Dynamic bus allocation
1	0 0 0 0 1	Initialize terminal
2	0 0 0 1 0	Transmit status word
3	0 0 0 1 1	Initiate self-test*
4	0 0 1 0 0	Initiate processing
5	0 0 1 0 1	Disable bus A
6	0 0 1 1 0	Disable bus B
7	0 0 1 1 1	Not used
8	0 1 0 0 0	Not used
9	0 1 0 0 1	Not used
10	0 1 0 1 0	Not used
11	0 1 0 1 1	Not used
12	0 1 1 0 0	Inhibit status word T/F flag (bit 0)
13	0 1 1 0 1	Reset status word T/F flag (bit 0) inhibit
14	0 1 1 1 0	Reset power off/on transient indicator
15-19		Spare
20	1 0 1 0 0	Initialize IPL
21	1 0 1 0 1	Read IPL record
22-31		Spare

*This command will initialize the RT prior to starting the self-test operation.

Figure 6.2-4. Command Word Definition



Where:

Message errors field is a 5-bit field consisting of the following message error flags:

Bit 8 INVALID CONTROL WORD
Bit 9 MULTIMESSAGE TRANSFER WORD COUNT ERROR
Bit 10 VALIDITY ERROR
Bit 11 COUNT ERROR
Bit 12 INVALID COMMAND

Subsystem faults (A) and (B) field will be defined by the individual equipment specification.

Figure 6.2-5. Remote Terminal Bit Word

6.2.3 System Control

The program load and data extraction channel (1553A channel 1) is not used to transfer operational data. The avionics system channel (1553A channel 0) provides system integration and operational data transfer. During operation, AN/AYK-14 standard airborne computer 1 acts as bus controller. If SAC-1 fails, bus control is assumed by SAC-2, another AN/AYK-14 with a slightly different configuration, which is normally used by the ASST subsystem to process electronic warfare data.

6.2.3.1 Fault Isolation and Redundancy Management

Communication status is established by the bus controller through periodic polling of each terminal. Polling occurs at the minor frame rate of 25 Hz. Because neither the data bus nor the subsystems contain redundant elements, redundancy management is not a function of the bus controller. BIT outputs from the primary controller are transmitted to the system control panel via the discrete data channel (channel 15).

6.2.3.2 Primary and Backup Control

AN/AYK-14 (SAC-1) normally serves as the bus controller. SAC-1 abnormal conditions are indicated by status lights on the system control panel. Operator intervention is then required to boot load the AOP into SAC-2 from the MTM via the program load channel (channel 1). The SAC-2 AOP is then manually initialized and SAC-1 powered down. SAC-2 status is also [illegible] discrete channel for display.

Because SAC-2 is normally used by the ASST subsystem, electronic warfare sensor data cannot be processed when the system is operating in the backup mode.

6.2.4 Bus Controller

Bus control for the LAMPS MK-III system resides within a Navy standard airborne computer AN/AYK-14(V). The AN/AYK-14(V) is a variable configuration, general-purpose, 16-bit computer featuring a performance range from 400 to 800 KOPS. The computer features a high degree of functional and mechanical modularity and is designed for flexible growth and extensive hardware commonality over a wide range of applications. The AN/AYK-14(V) architecture discernible to the user is not changed by modular hardware configuration changes, permitting common firmware and support software systems for all users.

The AN/AYK-14(V) consists of a family of plug in modules, chassis, interconnecting buses, support equipment, software, firmware, documentation, and training necessary to provide the user with a completely supported computer system. Currently, the AN/AYK-14(V) computer system provides 16 module types that can be configured in various combinations in three different chassis types. Configurations range from a two-module dedicated processor to multiple-processor computers with up to 524,288 words of memory and, with additional chassis, up to 16 I/O channels of various types.

LAMPS bus control is provided by a standard AN/AYK-14(V) module called the serial interface module (SIM). The SIM implements a serial multiplex data channel meeting the channel control and format requirements of 1553A. The module can operate with any 1553A protocol and can function as either a bus controller or remote terminal unit. Information is transferred on a single shielded-twisted pair line at a 1-MHz bit rate. Data are transferred in 20-us frames, each divided into one 3-us sync interval and 17 bit times of 1-us. All messages are addressed and use three types of words, command, status, and data per 1553A. Up to 32 terminals can interface on a single bus. All transmissions are initiated and controlled by the bus controller using standard message structures. Each SIM contains two 1553 multiplex data receiver-transmitters and sufficient logic and circuitry as necessary to detect, process, and participate in communication on two data buses.

Each AN/AYK-14 (primary and secondary) in the LAMPS system contain two SIMS, one for channel 0 and one for channel 1. Only one bus port is used for channel 0. since the avionics data bus is nonredundant.

Bus control is normally provided by SAC-1. This central system computer also provides the computation and control required to perform all mission-related airborne functions and to integrate the avionic equipment. It also communicates with the UYS-1 analyzer detecting set over a Proteus digital channel (PDC). It does not perform any signal processing; all such processing is done at the subsystem level. Backup bus control is provided by SAC-2 as described previously.

6.2.4.1 Primary Bus Control Implementation

Primary bus control resides within the AOP, which is loaded into SAC-1 from the MTM via 1553A program load channel 1. The AOP contains an algorithm

that controls the periodic communication between the computer and the RT's. Table 6.2-1 shows the polling frequency and I/O rate for each of the RT's. It can be seen from table 6.2-1 that each device on channel 0 has a polling rate that is a multiple of 40-ms. Each 40-ms cycle is subdivided into a polling period and a data transfer period. The polling period occurs at the start of the cycle and involves transmission of a transmit status word command to each RT scheduled to be accessed during this particular 40-ms cycle. During the polling sequence, the RT status word will be examined for the state of bits controlling subsequent data transfers to and/or from each RT. After completion of polling operations, data transfer will commence. During the data transfer period, each RT scheduled for data transfer activity will be commanded to send and/or receive data from SAC-1. After completion of all data transfer activity, the data bus will remain in a quiet state until the start of the next 40-ms cycle.

The number of data transfer commands is thereby reduced to the number of remote terminals that actually have data available for transmission. Simulation studies performed by the prime contractor indicate that the

Table 6.2-1. LAMP MK-III Bus Controller and Remote Terminal Data Rates

1553A channel 0			Maximum transfer			
			Terminal to controller		Controller to terminal	
Terminal address	Equipment connected	Polling rate (Hz)	Data words	Period (ms)	Data words	Period (ms)
00001	RTS (data link)	25	32	40	32	40
00010	EWS (electronic warfare)	25	32	40	32	40
00100	Signal data converter	12.5	2	1,000	4	80
00111	CSCG (communications)	5	29	1,000	17	200
01000	RHS (Doppler)	5	3	200	N/A	N/A
01011	Converter multiplexer	25	31	200	256	40
01110	NSIU (navigation)	5	2	1,000	6	200
10000	ATO control indicator (keyset)	5	2	200	10	200
11001	Armament SDC (ordnance)	12.5	2	80	1	80
11100	SD control indicator	5	2	200	10	200
As-required 1553 channel 1 (bus A or bus B)						
11111	Magnetic tape memory	N/A	2,000	1,000	2,000	1,000

average data bus use is 17% of capacity, and the peak load is 29% of capacity. The variation in bus traffic exists because of the asynchronous operation of the system, where only the data requested is transmitted rather than synchronous transmission of all the available data each minor frame.

6.2.4.2 Secondary Bus Controller Implementation

Secondary or backup bus control is provided by a second AN/AYK-14 of a slightly different configuration (SAC-2) that is normally used by the ASST system to process electronic warfare data. Upon failure or power down of the primary bus controller, the AOP (or a modified version) is loaded into SAC-2 from the MTM via program load channel 1. The AOP is then initialized and SAC-2 assumes bus control. It is expected that operator intervention will be required to effect the primary-to-secondary bus control transfer.
Bus operation under secondary control is essentially the same as that under primary control, except that electronic warfare signals can no longer be processed.

6.2.5 Remote Terminal

The LAMPS MK-III avionic data bus (channel 0) provides complete communication with seven major subsystems served by 10 RT's, as listed in section 6.2.1. The program load channel (channel 1) interfaces only with the MTM and the two AN/AYK-14 computers.

6.2.5.1 Subsystem Interface

All LAMPS data bus interfaces are integral to the subsystems that they serve. This approach simplifies management of interfaces and permits clear definition of responsibility for subsystem performance. The supplier of a subsystem is responsible for the proper operation of that subsystem on the data bus. The standard SIM contained within SAC-2 permits that computer to act as an RT when SAC-1 is controlling the bus or as a bus controller in the backup mode.

6.2.5.2 Fault Isolation and Redundancy

Although redundant data buses are used for the program load channel, the LAMPS avionic bus is nonredundant. A single 1553A data bus is used and the RT's contain no redundant elements.

The lack of redundancy in the RT's is partially compensated for by the use of extensive fault isolation and error detection and recovery techniques. Control of a faulty RT is accomplished by the transmission of a master clear signal on the AN/AYK-14 discrete channel. Master clear discretes may be either short pulse or long pulse. A short pulse provides a momentary clear signal to the RT. A long pulse or level can be used to hold an RT transmitter indefinitely in the off state. This signal may be used to recover from a fault condition that causes an RT to transmit continuously.

6.2.5.3 Error Processing and Error Recovery

LAMPS system errors may be detected either by the bus controller or an RT. Errors detected by the bus controller consist of status word timeouts and transmission faults. A status word timeout occurs if the RT does not

respond within a 7-us window provided for status word returns. A transmission fault occurs if the controller detects a data validity or word count error while receiving data or status words from an RT.

LAMPS RT status word definition is shown in figure 6.2-6. The message error flag, the terminal flag, and the high-temperature flag bits may be set in the RT status word to indicate RT-detected errors or faults. The presence of a logic 1 in any of bits 9-19 will cause an interrupt in the AN/AYK-14 CPU. Bits 10-18 may be masked if desired by the I/O controller.

Upon detection of an error, the 1553A channel controller generates an interrupt that halts I/O chaining activity. The operational program recognizes the interrupt and forms an error response based on the latest command sent to a terminal and the terminal BIT status word (see sec. 6.2.2.4). The error processing program stores a notification of each error occurrence identifying the error, RT, specific sequence, and error detection point involved. The error response is initiated in an out-of-sequence manner and will continue the previously defined I/O chain after recovery from the error.

If the recovery message is not successful (message errors or additional interrupts occur), a sequence failure notice is issued to the executive program and the I/O chain is entered at the start of the next logical sequence. A transmit BIT command is sent to an RT after the occurrence of a sequence failure to clear residual errors in the BIT word. The detailed error response sequence is specified by the prime contractor for each 1553A operation.

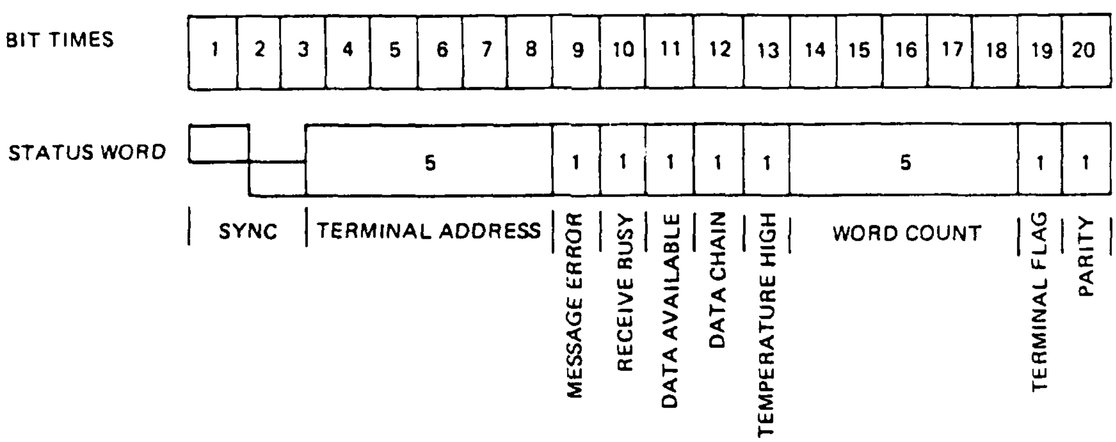


Figure 6.2-6. Remote Terminal Status Word Definition

6.3 YAH-64 ADVANCED ATTACK HELICOPTER MULTIPLEX SYSTEM -EXAMPLE 3

6.3.1 Application Area

The YAH-64 advanced attack helicopter (AAH) is being developed by Hughes Helicopters for the U.S. Army. It is tailored specifically for the day and night adverse weather antiarmor mission. A 1553A multiplex data bus system provides avionics, stores management, weapon fire control, and cockpit control and display integration.

In addition to its present applications, the AAH multiplex system may be extended to include other applications. Provision is being included for addition of the integrated avionics control set (IACS) to the AAH. In the future, certain information required by the flight control system may be routed over the data bus from avionic sensors to the fire control computer (FCC). From the FCC, these data would be forwarded to the flight control system either over dedicated lines or to a 1553A RT interfaced to the flight control system.

The AAH multiplex system can transmit a wide variety of signals. It replaces much of the signal and control wire and relay logic required in conventional aircraft configurations. Signals such as serial bidirectional, ac input, ac output, dc input, dc output, 28V discrete input, 28V discrete output, 5V discrete input, 5V discrete output, switch closure, synchro (three-wire) and ARINC 582-4 serial can be transmitted by the system.

Subsystems receiving and/or transmitting data via the multiplex data bus are as follows:

a. Fire control computer
b. Doppler
c. Target acquisition and designation system (TADS)
d. Pilot night vision system (PNVS)
e. Integrated helmet and display sight subsystem (IHADSS)
f. Hellfire missile control
g. Gun control
h. Rocket control
i. Electronic attitude director indicator (EADI)
j. Symbol generator
k. Heading and attitude reference system (HARS)
l. Fault detection and location system (FD/LS)
m. Mission equipment

Certain unique requirements were imposed on the multiplex system design. Survivability of the multiplex function in spite of battle damage to the aircraft was one such requirement. The issue of survivability will be addressed when describing the system architecture.

6.3.2 System Architecture

The AAH multiplex system is a 1553A time-division multiplex system consisting of 13 units that interface directly to dual redundant data buses. These 13 units process more than 1,300 signals. Of the 13 units, 9 are RT's specifically designed to adapt subsystems to the multiplex data bus. Where possible, interfaces to RT units have been standardized as

discrete (bilevel), ac and dc analog, synchro, and serial digital data. The multiplex system can be expanded to include 32 units to meet future requirements.

Figure 6.3-1 is a block diagram of the present AAH multiplex system. The system consists of:

a. Dual redundant data buses

b. Two bus controllers (primary residing in the fire control computer; backup residing in the copilot compartment remote terminal unit)

c. Symbol generator

d. Missile system remote electronics unit

e. Electronic attitude director indicator (EADI) remote electronics unit

f. Four general-purpose remote terminal units located in the pilot's compartment, right forward avionics bay, left forward avionics bay, and aft avionics bay

g. Four pylon remote terminal units (one located in each pylon)

h. Twenty-six data link termination units

Figure 6.3-2 illustrates the AAH multiplex system unit location in the aircraft. The primary data bus is routed along the left side of the aircraft, while the secondary data bus is routed along the right side. This isolation between the buses increases survivability. The two bus controllers (FCC and backup bus controllers) are widely separated for the same reason. Critical signals can be routed into separate RT units by providing separate signal paths, precluding the loss of that function because of an RT malfunction.

The AAH multiplex system generally complies with 1553A.

Redundancy is achieved in transmission lines, bus controllers, and RT's. Transmission line redundancy is provided by the use of dual redundant data buses in an active and standby arrangement. Two bus controllers are in the system. The primary bus controller resides in the FCC. The backup bus controller (BBC) is part of the copilot-gunner RT. The two bus controllers operate in a control and monitor fashion. In the RT's, redundancy is provided by dual modems and some duality of message decoding circuitry.

The multiplex system data bus operates asynchronously in a command/response mode, with transmissions occurring in a half-duplex manner. Each of the multiplex system data buses consists of a low-loss, twisted-shielded, 24-gage, Teflon-insulated wire pair, terminated at each end with its characteristic impedance (71 ohms nominal). All connections to the data bus system use a small (0.1 lb) data link termination (DLT) unit as illustrated in figure 6.3-3. The DLT units provide the bus with short-circuit isolation, impedance matching, and line termination. One DLT unit per bus is used for each terminal, thereby requiring two DLT units per terminal for the dual redundant system.

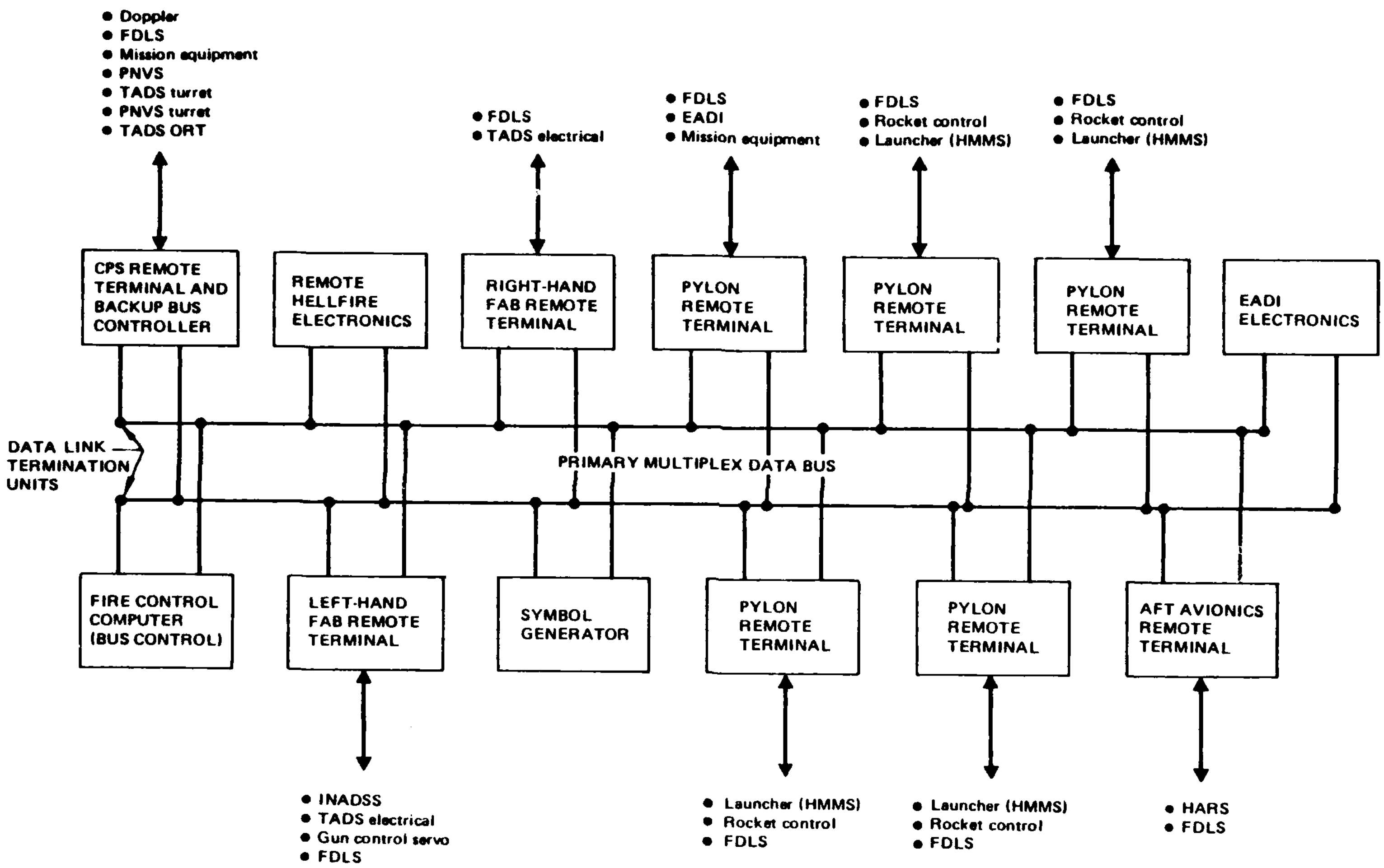


Figure 6.3-1. AAH Multiplex System Architecture

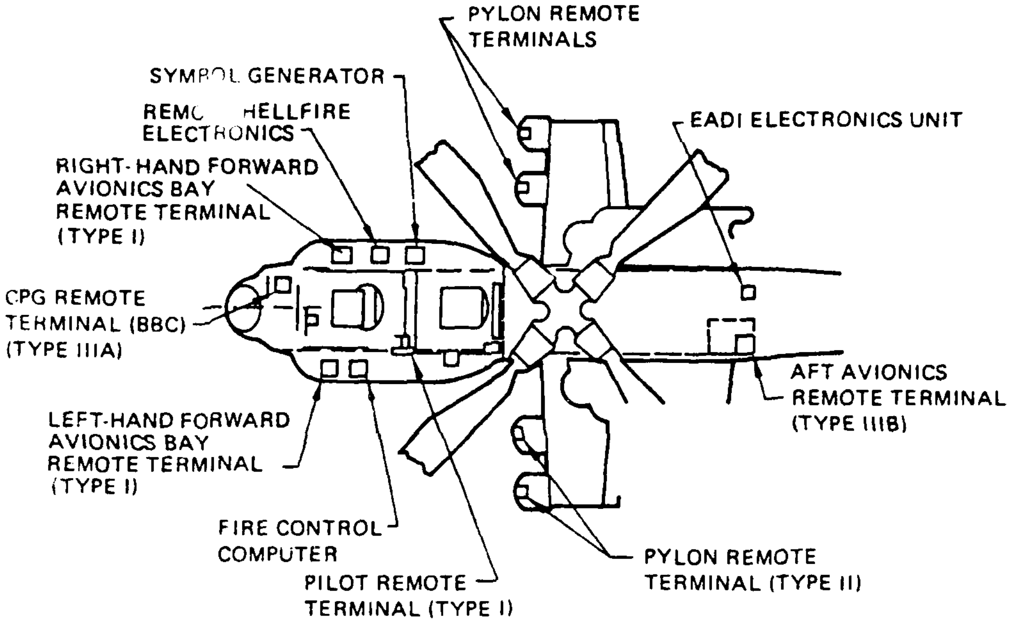


Figure 6.3-2. AAH Multiplex System Unit Location

Information flow on the data bus is composed of messages and words per 1553A. Data bus messages consist of controller-to-remote transfers, remote-to-controller transfers, and, although not presently used, remote-to-remote transfers. Data bus main frame time is a nominal 20 ms, as shown in figure 6.3-4. During this time, the active bus controller collects data from all boxes on the data bus (via transmit commands), performs the required logic processing and computations, and outputs the revised data to all boxes on the data bus (via receive commands). For subsystems that do not require this high update rate (50 times per second), data are processed and outputed at a lower rate (25 times per second).

The AAH uses command, data, and status words per 1553A. The command word is composed of a sync waveform, an RT address, transmit/receive bit, subaddress/mode field, data word count/mode code, and a parity bit. Data words consist of a sync waveform, 16-bits of data, and a parity bit. As implemented in the AAH, the status word is very unique. MIL-STD-1553A defines the status word as having a sync waveform, the RT address, a message error bit (M/E), a 9-bit status field, a T/F bit, and a parity bit. The status field of the status word is defined by the system designer. The AAH uses these bits as shown in figure 6.3-5 and as explained in table 6.3-1.

In addition to the normal bus control mode, the AAH data bus system is capable of operating in a dynamic bus allocation mode. The BBC was designed to recognize mode commands with the subaddress field of 00000. Subsequently, four mode codes are decoded with the 00000 code, resulting in a computer interrupt in the backup controller. With appropriate computer programming, the all 0's mode command is thus associated with the dynamic

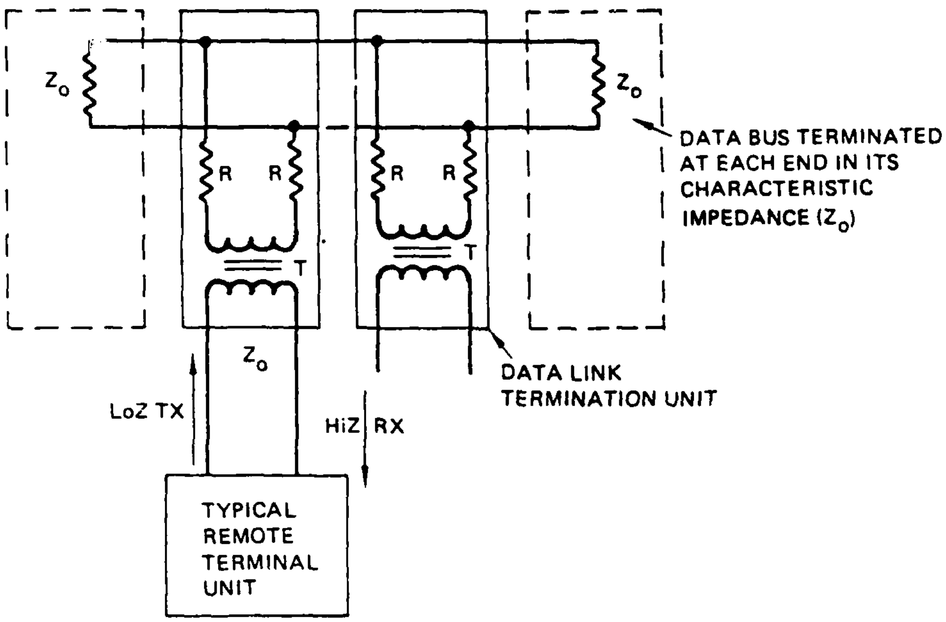


Figure 6.3-3. Lossless-Line Data Bus Configuration

bus allocation function. The three other decoded mode codes are assigned in the present AAH backup controller, as shown in table 6.3-2.

6.3.3 System Control

6.3.3.1 Fault Isolation

The strategy for retry and message list construction after a message transmission failure is a simple one. The controller will always use the channel that worked last for that message. For example, a successful transfer on bus A would result in the next transfer of the same message also being attempted on bus A; however, if the first attempt on bus A failed, then the retry of that transmission would be attempted on bus B. Thus, communications will continue on whichever channel is functioning. Note that the retry is limited to once per scan of the message list and is always initiated on the alternate bus. If the retry fails, the message is skipped. The sequence is "fail once, retry on alternate; fail twice, go to next message."

6.3.3.2 Primary and Backup Control

During normal operation, sole control of information transmission on the bus resides with the active bus controller, which initiates all transmissions. The primary bus controller resides within the FCC, while the BBC resides in the RT unit located in the copilot's compartment.

All data flow is controlled by addressed command words from the active bus controller to each multiplex RT unit, the symbol generator, the missile remote electronic unit, and the EADI electronic unit.

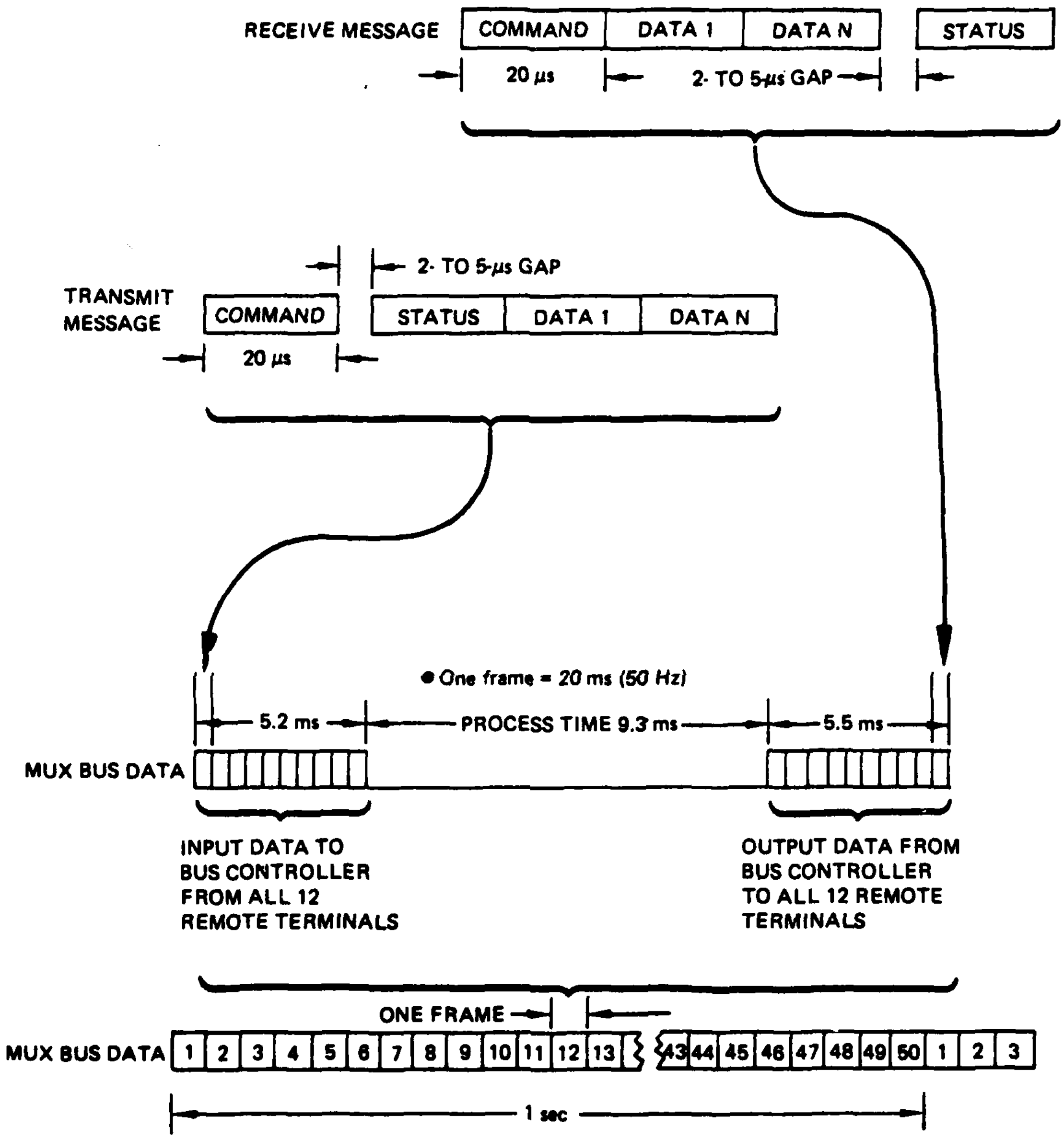


Figure 6.3-4. Data Bus Main Frame

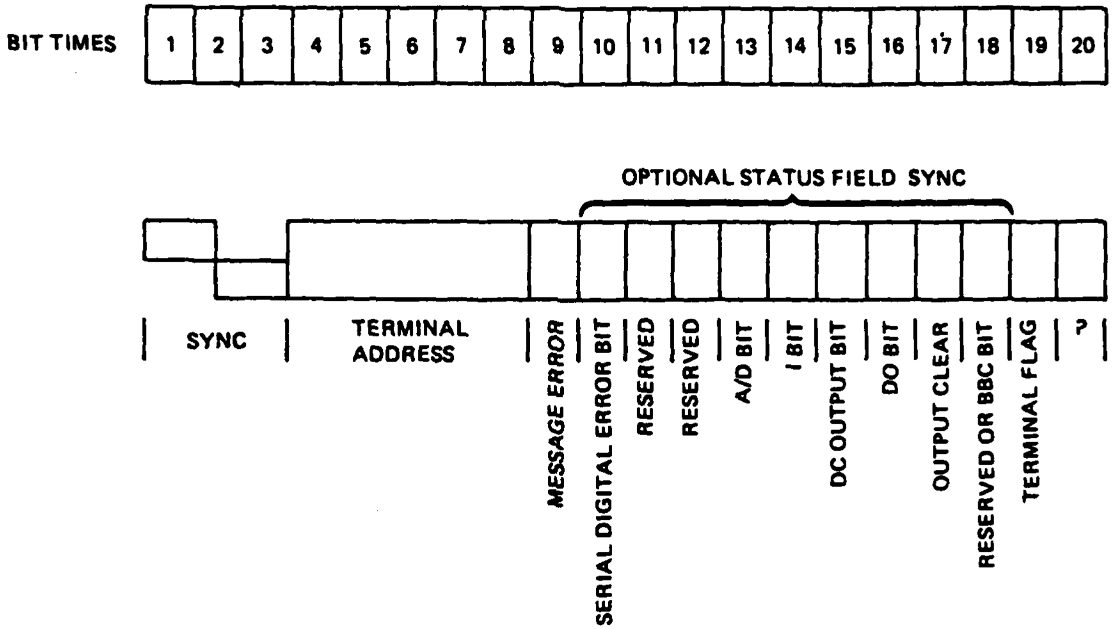


Figure 6.3-5. AAH 1553A Status Word

When the FCC is active and controlling the system, the BBC monitors bus activity. The BBC automatically assumes control of the system if the hardwired FCC signal to the BBC indicates a failure or if there is a loss of data transmissions on both data buses over a specific period of time.

If the FCC has relinquished control of the data bus to the BBC, it can automatically regain control of the system by indicating to the BBC via the hardwire connection that it is ready to resume normal operations.

If a faulty transmission occurs (command not properly executed), the active bus controller, based on data criticality, reissues the same command on the alternate data bus or initiates degraded modes.

When an RT unit connected to the data bus system is interrogated by the bus controller requesting data, the RT unit first sends a status word. This word provides the bus controller with the active and standby functional status of that unit. These data are used to help formulate degraded mode operation of the aircraft.

Either the primary or backup bus controller would normally be the active controller, depending on the status of handshake discretes, a selection switch in the copilot-gunner (CPG) cockpit, and activity of the buses. The block diagram of figure 6.3-6 shows the various signals associated with the selection of primary or backup bus controller.

Both controllers monitor the buses for activity in the form of valid commands. Lights in the CPG cockpit indicate whether primary or backup is

Table 6.3-1. AAH Status Word—Status Field Bit Definition

Bit time	Status bit
10	Serial digital error bit = used by RT's with serial digital interfaces to indicate failure of one (or more) of those interfaces
11	Reserved
12	Reserved
13	A/D bit = indicates failure of A/D converter(s) to pass BITE
14	I bit = indicates failure of discrete input-sampling circuitry to pass BITE performed at power up
15	DCO bit = indicates failure of dc output circuitry to pass BITE
16	DO bit = indicates failure of discrete output circuitry to pass BITE
17	Output clear = indicates outputs are being held in clear state
18	Reserved for all RT's except CPG remote terminal and backup bus controller
	BBC bit = for CPG terminal and backup bus controller, indicates that backup bus control computer is up and running

in control and also cue the CPG that the backup controller has detected an inactive bus when the primary was supposed to be in control. The logic for bus controller selection is given in table 6.3-3.

6.3.3.3 Fault Detection and Location Subsystem

Integrated within the YAH-64 multiplex subsystem is the fault detection and location system (FD/LS). Detection and isolation of electrical and electronic LRU failures uses the multiplex data bus system and associated hardware to perform various functions. The RT's are used for signal conditioning and data transfer. The active bus controller is used for fault processing, control, and data storage. The multifunction data entry keyboard located in the CPG compartment is used for FD/LS command initialization, and the TADS alphanumeric displays are used for fault location presentations. The present system fault-detects and isolates over 69 replaceable units.

Table 6.3-2. Mode Code Functions

Mode code	Function
00000	Issues computer interrupt in backup bus controller (dynamic bus allocation)
00001	Clear remote terminal
00010 and 11111	Decoded but not used
Others	Not decoded

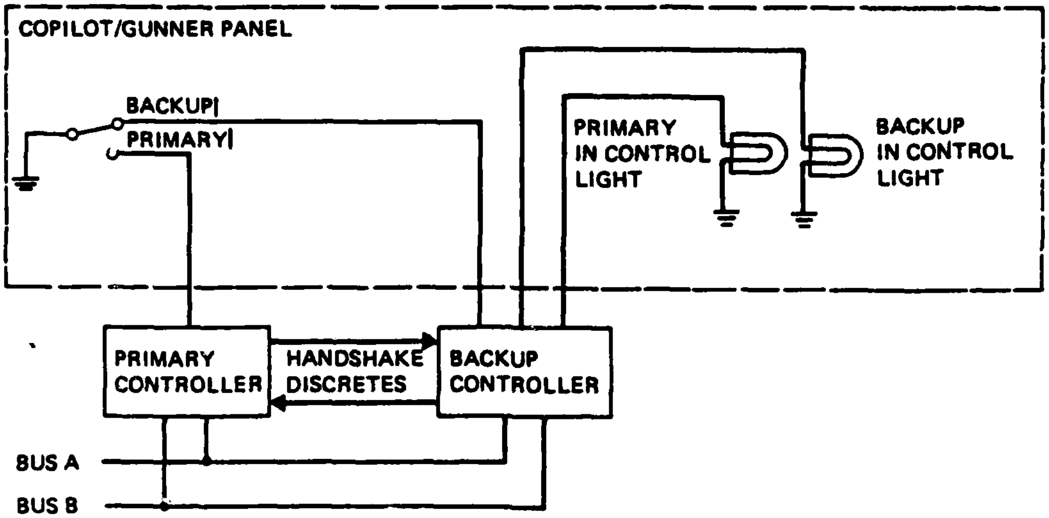


Figure 6.3-6. Primary and Backup Bus Controller Selection Logic

The copilot crew station layout showing the locations of the multifunction data entry keyboard and the TADS alphanumeric displays is shown in figure 6.3-7. The FD/LS is capable of correctly identifying a failed LRU in the air or on the ground for at least 95% of the failures monitored, weighted by failure rates. In addition, the system false alarm rate does not exceed 2%.

The active bus controller, operating from data received over the multiplex data bus and [illegible] equipment status stored in memory, processes programmed algorithms as required to determine the operational status of the monitored LRU. The bus controller outputs the required information to the symbol generator, which generates alphanumeric messages for display on the TADS alphanumeric display.

When in flight, the FD/LS continually and automatically monitors operational status of the AAH electronic and electrical subsystems. If a fault occurs, the aircrew is alerted so appropriate action can be taken. Major subsystems that are so monitored include the missile subsystem, rocket subsystem, gun, automatic stabilization equipment, pilot night vision, integrated helmet-mounted sight and display, Doppler, TADS, symbol generator, EADI, engine instruments, auxiliary power unit, and electrical subsystem.

Table 6.3-3. Bus Controller Selection Matrix

Controller	CPG switch	Handshake discretes	Bus activity
Primary to assume control	Primary	Backup not in control	None for 120 ms (programmable) or at power up
Backup to assume control	Primary	Primary not in control	None for 120 ms (programmable)
	Backup	Do not care	

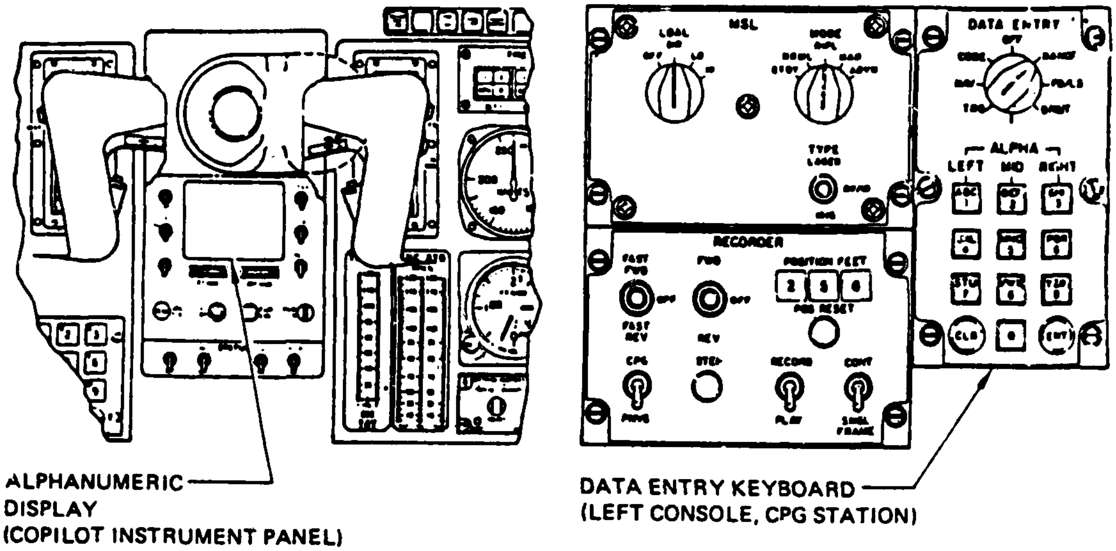


Figure 6.3-7. Fault Detection Location Subsystem Controls and Displays

Through special keyboard entries, either flight or maintenance personnel can command a particular subsystem fault detection and location test routine. At the completion of the test, the results are displayed on the TADS alphanumeric display for evaluation. In addition, in the maintenance mode, a complete aircraft end-to-end fault detection/location test can be commanded through a maintenance keyboard entry. This test will check all systems connected to FD/LS and will display all failed units on the TADS alphanumeric display.

6.3.4 Bus Controller

6.3.4.1 Primary Bus Controller

The primary bus controller resides in the FCC. The FCC is a MECA-43 16-bit-word hybridized microcomputer manufactured by Teledyne Systems. It is a general-purpose, microprogrammed, digital-parallel, synchronous machine with 16K words of ROM memory and 2K words of RAM memory.

The 1553A bus control interface was designed to enable the Teledyne FCC to function as bus controller and/or RT in a dual redundant multiplex data bus system (see fig. 6.3-8). The interface complies with 1553A requirements and is designed to connect directly to most general-purpose computers. For minimum size and weight, all circuit components are in dice form fabricated in hybrid packages. There are three functionally unique hybrids:

a. Driver-receiver hybrid
b. Multiplex terminal unit (MTU) hybrid
c. Device control unit (DCU) hybrid

The driver-receiver hybrid couples directly to a dual data bus system accommodating TTL control and data signals on one side and +10V Manchester

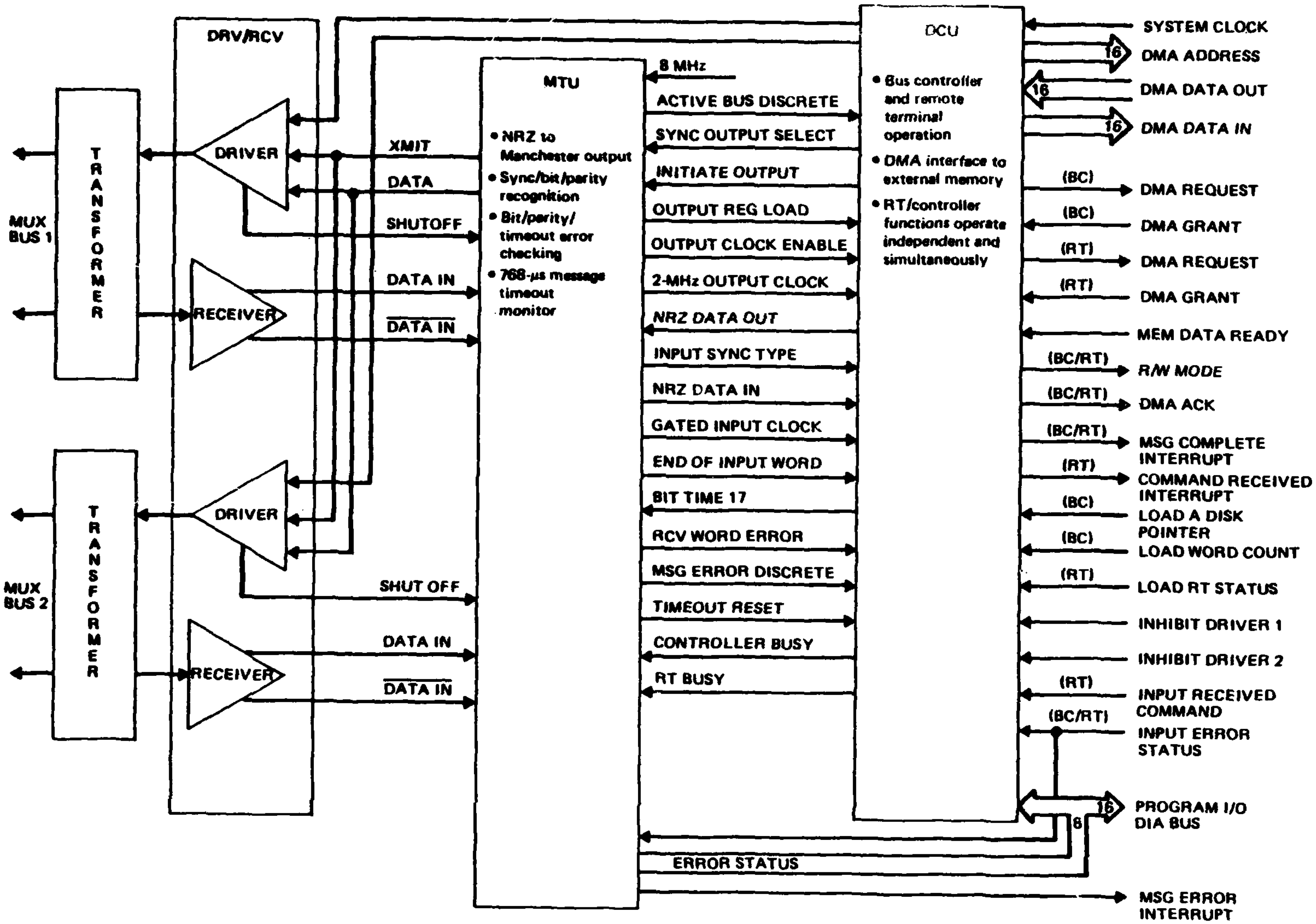


Figure 6.3-8. AAH 1553A Hybrid Bus Control Interface

biphase signals on the bus side. The MTU provides a full-duplex serial interface between the driver-receiver and DCU hybrids. MTU functions include code conversion between NRZ and Manchester, serial timing and formatting for I/O data, validity checks on received data, and a total message length monitor. The DCU performs all message-handling requirements of a bus controller and/or RT. It uses the computer's main memory for working storage moving data in and out via direct memory access (DMA).

Computer software presides over DCU bus controller operations. Each bus transaction is initiated by outputting a message pointer and a control word specifying the number of transmit-receive words. As an RT, the program specifies RT address and controls 10-bits of the status word. Controller and RT functions are assigned separate DMA ports, thereby enabling simultaneous bus controller and operation for 100% self-test capability. Other design features include--

a. Bus controller memory addressing up to 65K; separate RT memory addressing of 2K

b. A next message queue for continuous controller operations

c. RT-to-RT messages

d. A message error status register with interrupt and an automatic system reset on excessive message length times

e. A dead-line discrete indicating no bus traffic

f. In the RT mode, buffer holds the last received command for program interrogation

Device Control Unit

The DCU performs those functions required of a bus controller and RT as specified in 1553A for message processing. It is responsible for all message contents into and out of the local subsystem's own memory. The connected subsystem (computer or controller) directs the activities of the bus controller portion of the DCU by sending it starting addresses of the message in its memory and the number of words (commands and data) to be transmitted plus the number of words to be received (status and data). The RT portion of the DCU operates without subsystem intervention other than specifying the RT's address and 10 least significant bits of status.

The five most significant bits of the 16-bit RT DMA address select a 2,048 word block assigned to buffer messages when the DCU is operating as an RT. These bits are available on external pins. The DCU will operate independently and simultaneously as bus controller and/or RT, i.e., the bus controller can talk to itself as though it were 1 of 32 RT's providing a 100% closed-loop, self-test capability.

To initiate a message on the bus, the local subsystem outputs two control words to the DCU. One specifies a starting address in the subsystem's memory, and the other defines the number of words involved. These control word formats are shown in figure 6.3-9.

CONTROL WORD OUTPUT 1	16 BITS DMA STARTING ADDRESS POINTER					
CONTROL WORD OUTPUT 2	SEL BUS A	SEL BUS B	RT TO RT		6 bits = number of RCV words	6 bits = number of XMIT words

Figure 6.3-9. DCU Control Words

Multiplex Terminal Unit

The MTU interfaces the DCU to the bus drivers and receivers by performing all bit and word serialization on data to and from the bus. It accepts Manchester encoded data from the line receivers and outputs NRZ data to the DCU. Similarly, it takes (on command) serial NRZ encoded data from DCU and supplies Manchester encoded data to both line drivers. Input and output occurs independently and simultaneously, thus providing a full duplex serial interface to the DCU. The MTU also performs bit error checking, word parity checking, and a total message length time check. A dead-line discrete is also provided indicating no traffic on the bus for a time exceeding 65.5 ms.

Dual Driver-Receiver

The driver-receiver hybrid provides the interface between the MTU and one of two 1553A data buses. Bus switching is under program control. It consists of two sets with a driver section, a receiver section, isolation resistors, and coupling transformers (external to the hybrid).

6.3.4.2 Backup Bus Controller

Backup bus control is provided by a SDP-175 microprocessor designed and built by Sperry Flight Systems. The processor is located in the CPG RT unit. The 2901A 4-bit-slice microprocessor is a microprogrammed digital computer capable of performing a degraded mission function, as well as backup bus control, upon loss of the FCC. The SDP-175 has 12K words of ROM memory and 2K words of RAM memory.

The BBC is unique in that it is located in the same housing as an RT but is functionally separate from the RT. The functions are split between RT control and BBC in such a way that the RT cannot determine whether it is receiving a command from the primary or backup controller. Both the backup bus control computer and the RT control transmit their information on the data bus and respond as though receiving information from a source outside their own box. The functional separation of RT and bus control allows either one to operate in case the other one fails, barring failure of a physically shared component such as a power supply or the bus interface electronics.

Figure 6.3-10 presents a block diagram of the CPG RT and BBC unit.

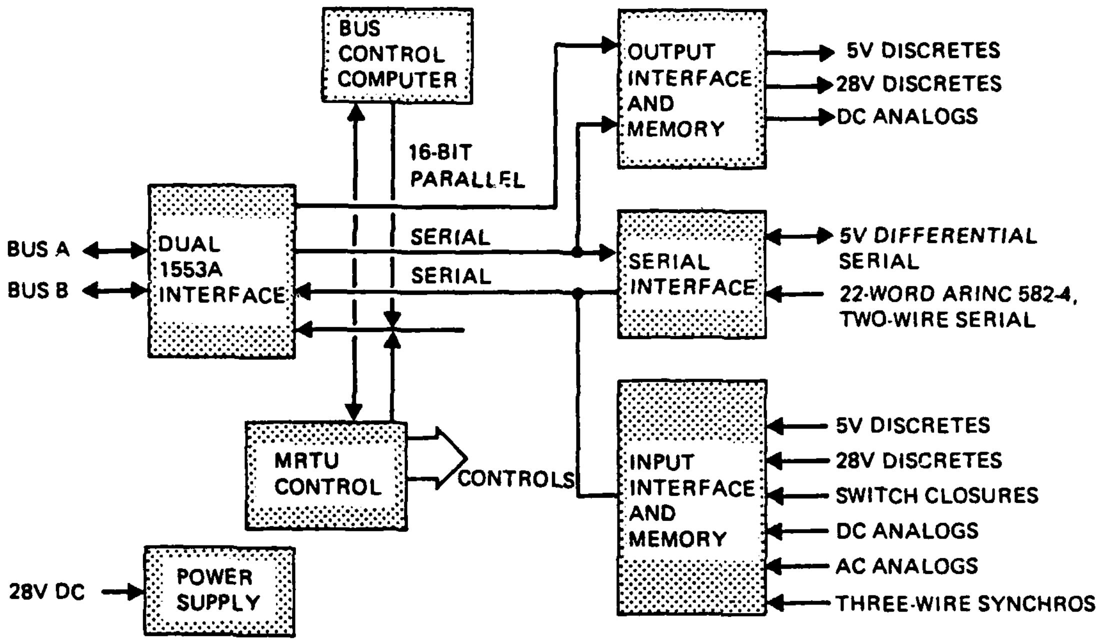


Figure 6.3-10. CPG Remote Terminal and Backup Bus Controller

6.3.5 Remote Terminal

As presented earlier, 13 units are connected to the multiplex data bus. Four of these units (FCC, remote hellfire electronics, symbol generator, and EADI electronics) have imbedded dual redundant 1553A interfaces. The other nine units are multiplex *remote terminal units (MRTU) built by Sperry.*

The RT units (identified as types I, II, and III) input and output a standard assortment of bilevels, ac and dc analog, serial digital, and synchro I/O signals to all parts of the aircraft. To fit the needs of the YAH-64 aircraft, these units contain different I/O signal capacities as represented in table 6.3-4.

All RT units contain dual redundant 1553A data bus interfaces. When further redundancy is required, such as for a critical input or output signal, that particular signal is wired into two separate RT units.

Each RT unit contains sufficient BIT circuitry to detect 95% of all faults (weighted by failure rate) within itself. At the LRU level, these units contain an internal test system to check input and output channels for integrity as well as power supplies and other internal circuitry. A hardware timeout function is provided in the RT, to shut off a continuous transmission by a faulty transmitter.

Table 6.3-4 also shows the physical outlines, size, weight, and power of the three types of multiplex RT's. The three types of RT's have been designed for hard mounting without shock absorbers, conventional cooling,

Table 6.3-4. Multiplex Remote Terminal Unit Specifications

	MRTU I	MRTU II	MRTU III
Dimensions (H x W x D) (in)	5 x 7 x 7.42	3.1 x 4.28 x 8.8	5 x 7 x 10.25
Weight (lb)	9	4	12.5
Power dissipation (W)	25	10	40
Typical signal mix			
Serial bidirectional	4	2	4
Ac input	4	–	4
Ac output	8	–	–
Dc input	20	8	20
Dc output	20	4	20
28V discrete input	16	–	16
28V discrete output	16	8	16
5V discrete input	48	–	48
5V discrete output	56	–	56
Switch closure	56	–	56
Synchro (three wire)	–	–	4
ARINC 582-4	–	–	1
Total signal count	250	22	245
Bus controller	No	No	Yes

100% plug-in modules (except power supply), and handmated connectors. These Sperry MRTU's use hybrid device technology to implement the RT electronics.

The type II RT that is mounted in the AAH pylons requires an unusual box envelope. Card commonality between all types of RT units is maintained in spite of the unique packaging requirements for the type II RT unit. The cards are mounted vertically in the types I and III RT's, and are mounted horizontally in the type II MRTU.

A unique feature of the RT's is that MRTU type III contains the SDP-175 microprocessor. As presented previously during the discussion of backup bus control, the SDP-175 serves the AAH as a backup mission computer and, in the multiplex system, as a bus monitor and backup bus controller. The functional separation of the RT and backup bus control functions of MRTU type III is noteworthy.

6.4 B-52 OFFENSIVE AVIONIC SYSTEM MULTIPLEX SYSTEM -EXAMPLE 4

6.4.1 Application Area

The B-52 offensive avionics system (OAS) is a retrofit update to the B-52 avionics being incorporated to improve mission reliability, reduce life cycle cost, and support the air-launched cruise missile (ALCM) weapons system. The OAS uses 1553A data buses as its information transfer medium. The application areas of the multiplex system are navigation, stores management, and control and display. The multiplex system uses two active and standby pairs of data buses.

The OAS data processing is basically centralized. The data bus traffic includes inertial platform data, missile alignment data, and all (except safety related) control and display data. Some safety aspects excluded from the multiplex buses relate to nuclear safety, as the B-52 OAS controls, monitors, and delivers nuclear weapons. Nuclear safety and survivability requirements imposed on the OAS are probably unique to strategic aircraft and their systems. For example, the OAS must remain operational during and after a nuclear event. The B-52's navigation system is required to be self-contained, and the aircraft must not become "lost" because of any type of transient. These safety and survivability requirements, along with requirements for high weapon delivery accuracies, lead to subsystem requirements to store critical data in multiple locations and to recover rapidly from failures and upsets.

Subsystems receiving and/or transmitting data via the multiplex data bus are as follows:

a. Two avionics processors (AP)
b. Two inertial measurement units (IMU)
c. Doppler velocity sensor (DVS)
d. Autopilot
e. Attitude heading reference system (AHRS)
f. Air data elements
f. Four data transfer units (DTU)
g. Electro-optical viewing subsystem (EVS)
h. Angle-of-attack (AOA) computer
i. Advanced capability radar (ACR)
j. Control and display
k. Radar altimeter
l. Weapons control and delivery system

6.4.2 System Architecture

6.4.2.1 Physical Architecture

The physical architecture of the B-52 OAS multiplex system consists of four buses, twisted-shielded wire pairs terminated at both ends with the characteristic impedance of the wire pair, and 17 electronic units, each connected to two or all four of the buses. Two of the 17 units are avionics processors and are connected to all four buses. Nine units are connected to two of the buses, and six units are connected to the other two buses. This architecture provides two bus pairs with each avionics processor connected to both (see fig. 6.4-1).

The 17 units connected to the 1553A buses are as follows:

a. Navigation and weapons delivery avionics processor (NAWD-AP)
b. Control and display avionics processor (CAD-AP)
c. Inertial electronics unit #1 (IEU-#1)
d. Inertial electronics unit #2 (IEU-#2)
e. Radar interface unit (RIU)
f. EVS interface unit (EIU)
g. Armament interface unit (AIU)
h. Display electronics unit (DEU)
i. Doppler velocity sensor (DVS)

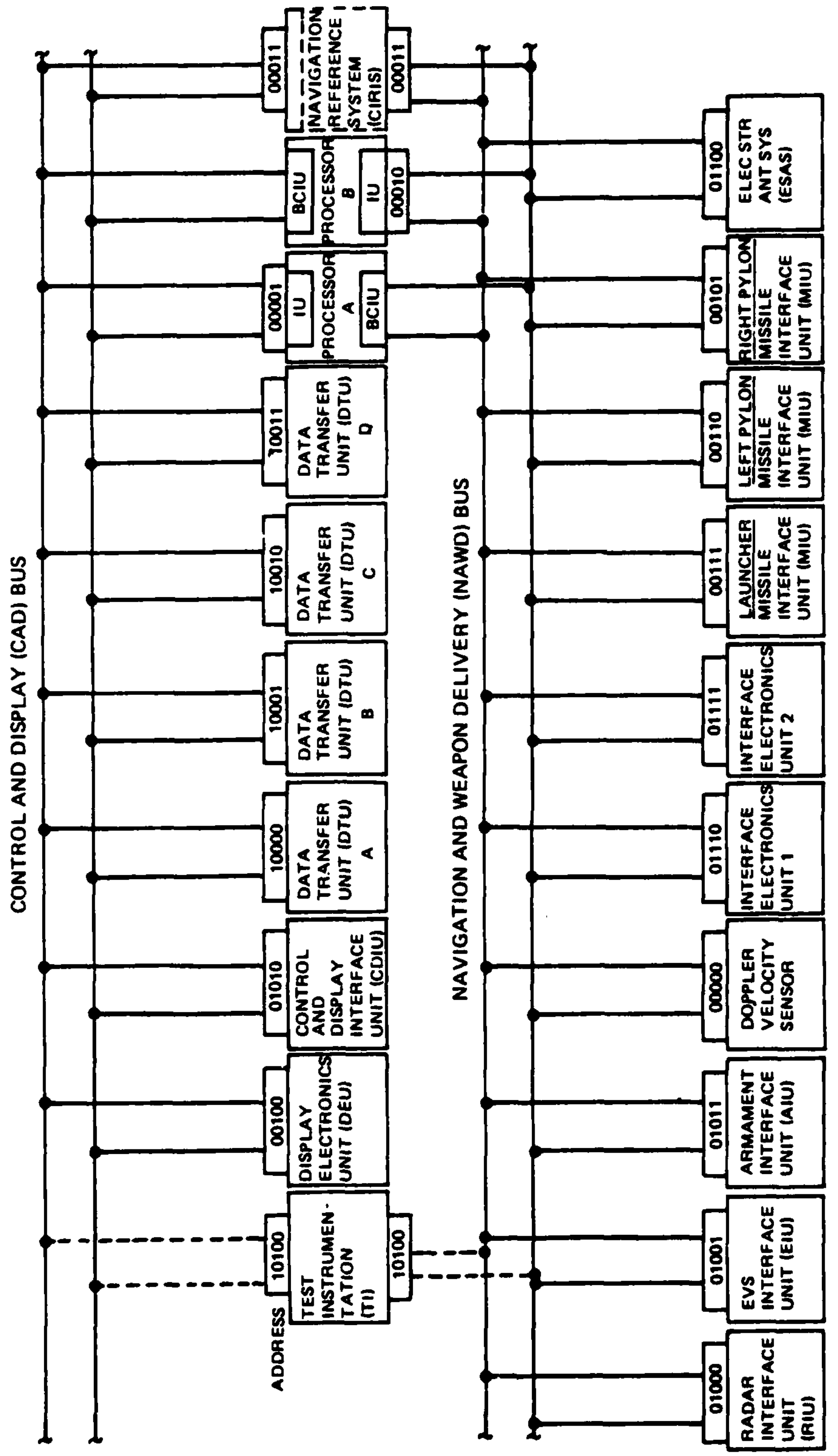


Figure 6.4-1. OAS Multiplex System Architecture

j. Control and display interface unit (CDIU)
k. Missile interface unit -- rotary launcher (MIU-RL)
l. Missile interface unit -- left pylon (MIU-LP)
m. Missile interface unit -- right pylon (MIU-RP)
n. Four data transfer units (DTU-A, DTU-B, DTU-C, and DTU-D)

6.4.2.2 Functional Architecture

Functionally, the B-52 OAS has a federated computational architecture that uses two processors. One processor performs navigation and missile processing, and the other processor performs controls and displays processing. In the event of a processor failure, the other processor has been specified to perform time critical and critical (but not noncritical functions) in a backup mode. Four 1553A buses are operating as two active and standby pairs, with each pair controlled by a separate processor.

The functional architecture is shown in figure 6.4-2. It is represented as directional message flow between the two processors and between RT's and processors.

6.4.2.3 Documentation and Conformance to 1553A

The system specification requires the use of 1553A data buses. A Boeing document, "B-52 OAS Multiplex Bus Protocol" (D675-10110-1), expands on the standard. All vendors and designers were required to use this document. RT-to-RT and broadcast transmissions are not used in the OAS, although RT-to-RT transmission is described in the protocol document.

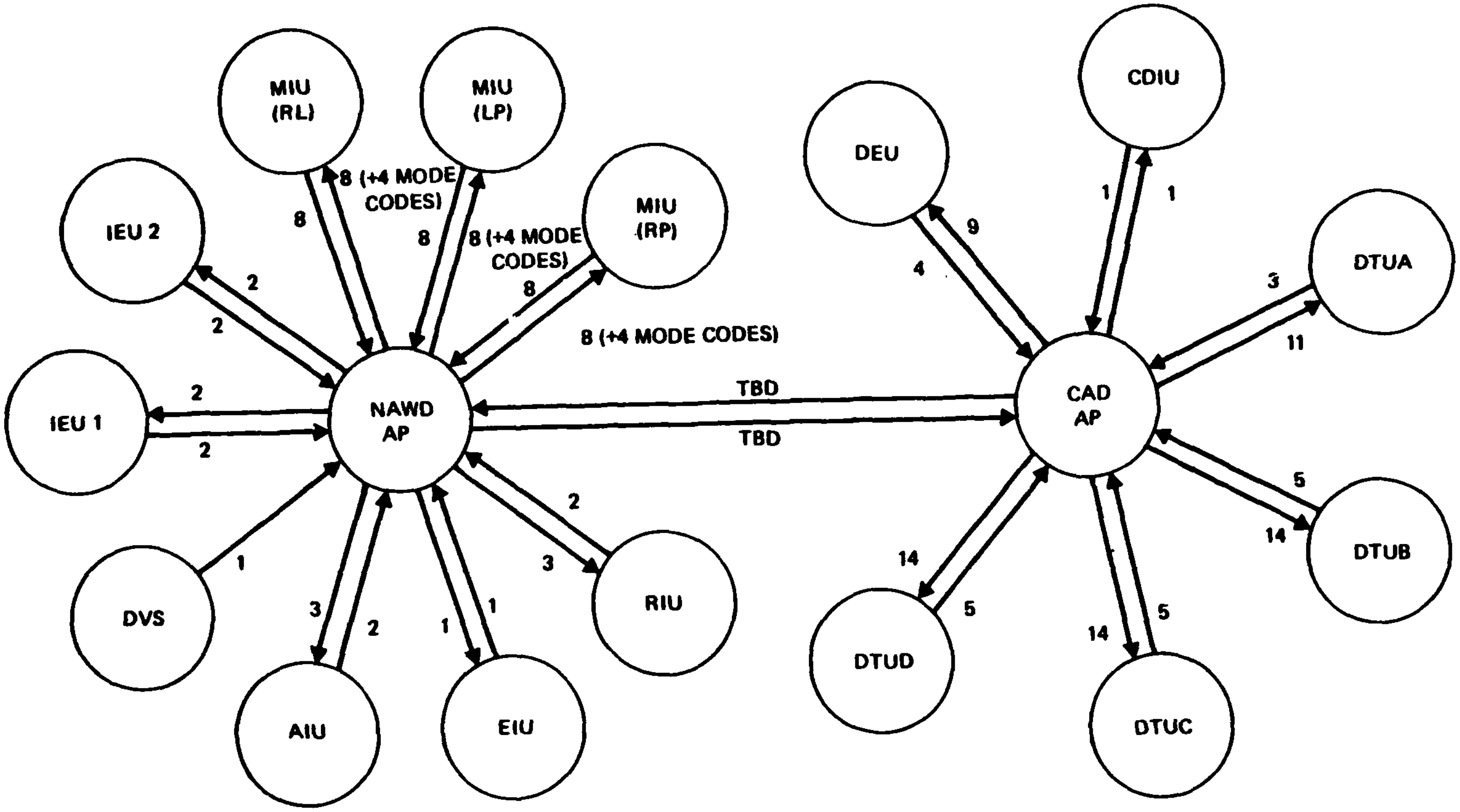
The OAS generally complies with 1553A. The status word and mode codes used in OAS comply with 1553A but are different from those required by 1553B.

6.4.2.4 Redundancy

Redundancy in the OAS is achieved in a variety of ways. Use of four buses provides redundancy of data paths, as well as load splitting, since they are used as two active and standby pairs. Two identical processors provide physical redundancy for backup mode operation, as well as partitioned functions for normal operation. Within each processor there are totally independent I/O channels for each of the four buses that provide necessary independence and redundancy. The RT's have redundant bus interface units (BIU). Some subsystem redundancy is provided in the OAS by two inertial measurement units, each connected to the bus through its own remote terminal IEU (see fig. 6.4-3).

6.4.2.5 Bus Network

At present, the OAS bus network is still under development, and the length of the buses and stubs has not been finalized. The length of the two buses on which the NAWD processor is master will be about 250 to 350 ft. Subsystem equipment relocation is being studied to reduce the length of these NAWD buses. The two buses controlled by the CAD processor are connected to subsystems concentrated in the cockpit and forward sections of the B-52. The CAD buses are expected to be relatively short, about 20 to 50 ft in length.



Note: Numbers indicate number of messages.

Figure 6.4-2. B-52 OAS Message Flow

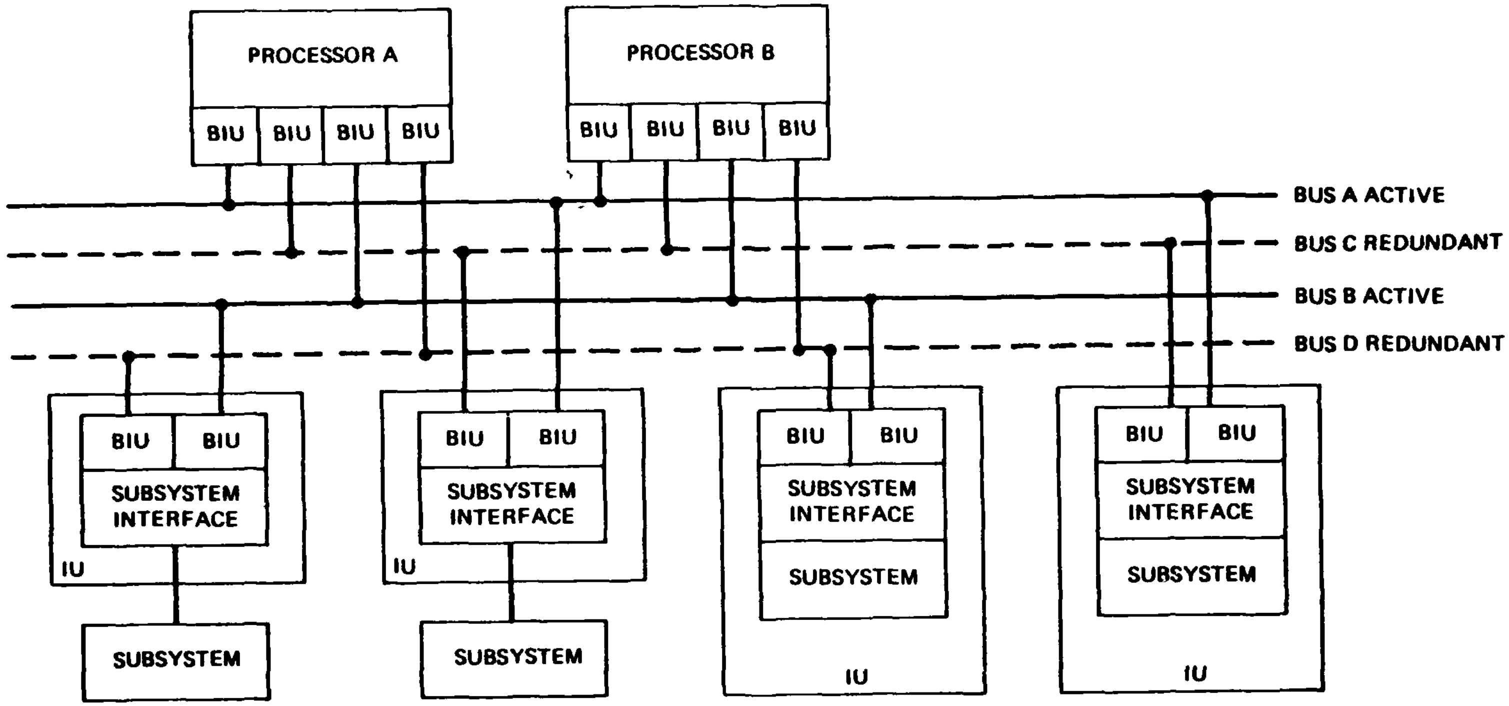


Figure 6.4-3. OAS Multiplex Redundancy

The OAS multiplex system will use transformer-coupled stubs, as shown in figure 6.4-4. Because of the placement of certain RT's, such as the MIU for the rotary launcher, one or more very long stubs may have to be used. The stub for the MIU-RL may be 40 ft long. To compensate for waveform distortion at the receiver of this RT, additional filtering of the received signal will probably be incorporated.

6.4.2.6 System Synchronization

System synchronization is achieved by using real-time clock interrupts. Each processor is capable of receiving two real-time clock interrupts, one from its own real-time clock and one from the other processor.

During initialization, one of the real-time clocks is selected to be master. This real-time clock interrupt deemed to be master is enabled in both processors and is used to ensure that both processors are properly synchronized. The master real-time clock interrupt routine passes control to the task scheduler. Each processor also monitors its own internal real-time clock to ensure that the master real-time clock has not failed. If the master real-time clock fails, control is passed to the operational computer program (OCP) initialization for reconfiguration. If a foreground processing function is running when the master real-time clock interrupt occurs, the minor frame overrun flag is set and the control returns to the interrupted foreground function.

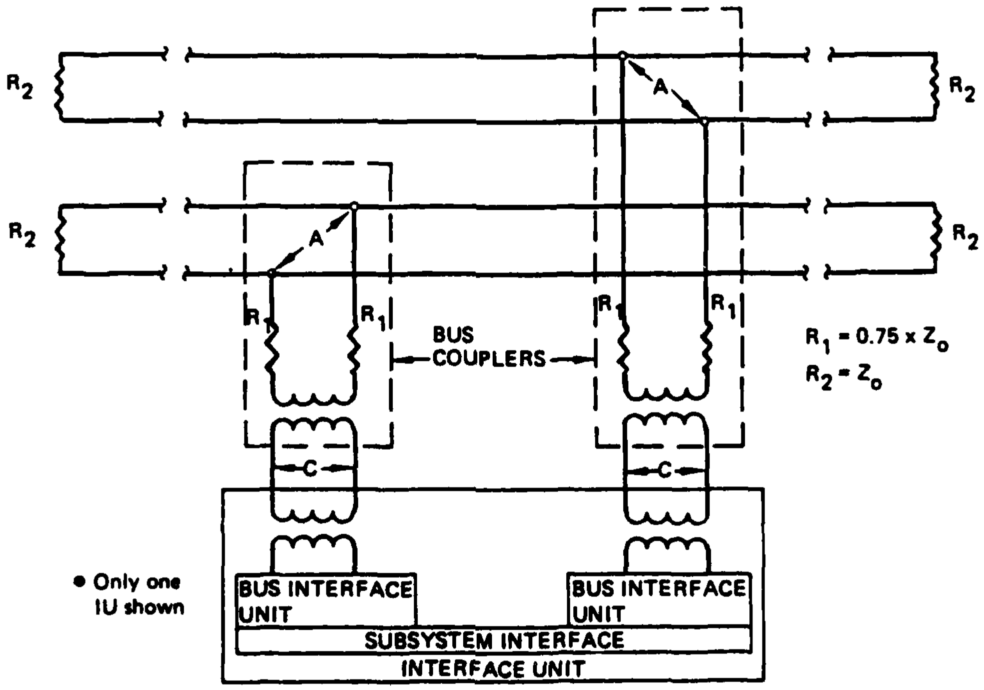


Figure 6.4-4. Data Bus Coupling

6.4.2.7 Bus Protocol

Bus protocol is per 1553A. All transactions are strictly command/response. Each AP is the bus controller on one pair of buses and an RT on the other bus pair. RT-to-controller and controller-to-RT data transmissions are the only ones that are implemented. Mode commands can be transmitted over the bus for the purpose of multiplex system management.

RT-to-controller and controller-to-RT transmissions are accomplished by 1553A command, data, and status words (see fig. 6.4-5). The command and data word formats are as specified in 1553A. The specific mode codes implemented in the OAS will be discussed later. The status word has a 9-bit status field that was specified by the system designers and is unique to the OAS.

The OAS status word begins with a 3-bit time invalid Manchester sync and ends with an odd-parity bit in bit time 20. As specified in 1553A, bit times 4 through 8 are the terminal address, bit time 9 is the message error bit, and bit time 19 is the terminal flag. Status word bit times 10 through 18 are the status field. The status field bit assignments are shown on figure 6.4-5 and further defined in table 6.4-1. The status field bit assignments are one of the two major areas of difference between the OAS 1553A and 1553B. A status word comparison is presented in figure 6.4-6.

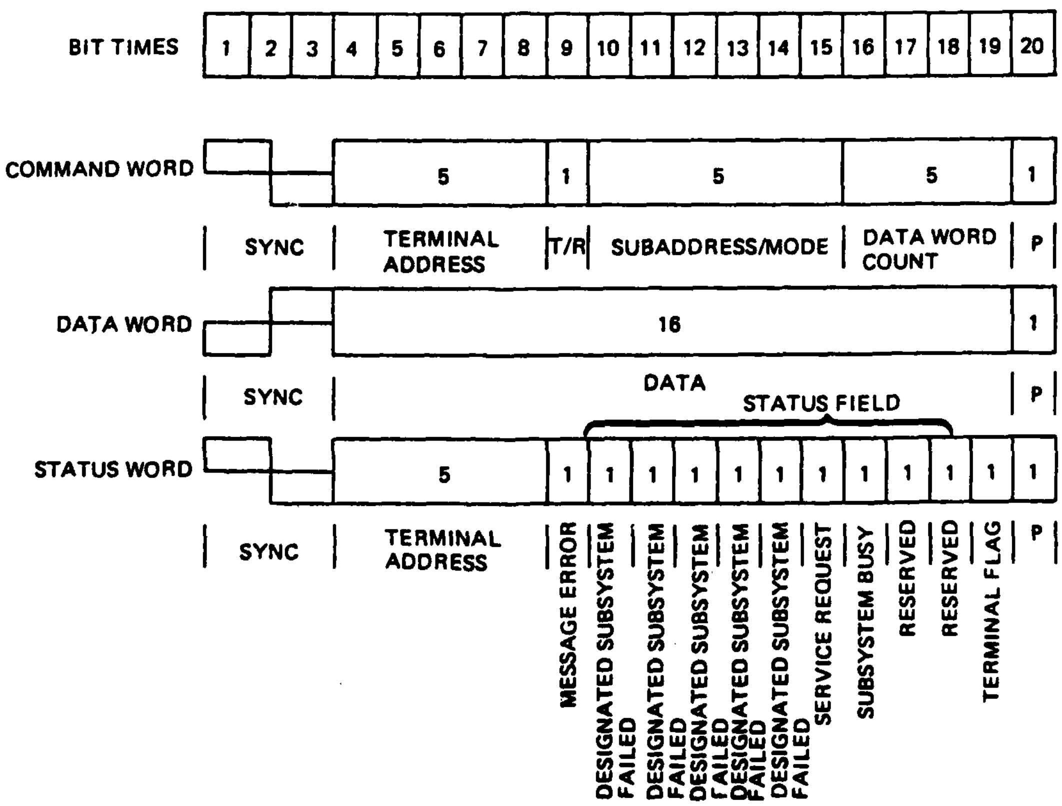


Figure 6.4-5. Multiplex Protocol Word Formats

Table 6.4-1. Status Field Bit Definition

Status code									Bit function
10	11	12	13	14	15	16	17	18	
x	x	x	x	x	x	x	x	0	Reserved
x	x	x	x	x	x	x	0	x	Reserved
x	x	x	x	x	x	1	x	x	Indication that it cannot supply data due to busy status of subsystem
x	x	x	x	x	1	x	x	x	Indicates RT request service
x	x	x	x	1	x	x	x	x	Indicates a designated subsystem failure
x	x	x	1	x	x	x	x	x	Indicates a designated subsystem failure
x	x	1	x	x	x	x	x	x	Indicates a designated subsystem failure
x	1	x	x	x	x	x	x	x	Indicates a designated subsystem failure
1	x	x	x	x	x	x	x	x	Indicates a designated subsystem failure

x—don't care

The OAS implements four mode codes for use in management of the multiplex system. Use of these mode codes is the second area of major difference between the OAS 1553A and 1553B. Table 6.4-2 compares 1553A, OAS definition, OAS RT implementation, and 1553B mode codes. Note that in 1553A only one mode code is defined, and all others are left for the system designer to define and use as required. In 1553B all mode codes are defined or reserved, and the system designer merely chooses and implements the mode codes for this particular system from those specified. It is the Air Force's intent in 1553B that bus controllers be able to generate all mode codes if mode codes are used.

6.4.2.8 Multiplex System Timing

Multiplex system timing is accomplished by implementation of specific major and minor frames. The OAS major frame has a frequency of 16 Hz. There are four minor frames per major frame, giving a minor frame frequency of 64 Hz. Each minor frame is initiated by a real-time clock interrupt, as explained previously under system synchronization.

6.4.3 System Control

6.4.3.1 Fault Isolation and Retry Scheme

Bus control is based on the ability to communicate. Communication status assessment is established by system software that interrogates each LRU at periodic intervals for its status on each bus. If communication is not attainable after three consecutive tries on one bus, then the operation is

Table 6.4-2. Multiplex Mode Code Comparison

Mode code (control word field) Bits					OAS mode code definition*	OAS RT implementation	1553B mode codes
15	16	17	18	19			
0	0	0	0	0	Reserved	None	Dynamic bus control
0	0	0	0	1	Transmit status word	All RT's	Synchronize
0	0	0	1	0	Initiate self-test	Optional	Transmit status word
0	0	0	1	1	Transmitter disable	All redundant RT's	Initiate self-test
0	0	1	0	0	Transmitter enable	All redundant RT's	Transmitter shutdown
0	0	1	0	1	Reserved	None	As specified in 1553B
↓		↓			↓	↓	↓
1	1	1	1	1	Reserved	None	As specified in 1553B

*1553A defines mode code 00000 as dynamic bus allocation; all other mode codes are undefined.

alerted and the failure is recorded for maintenance action. The periodic interrogation interval will be long enough to prevent a power system transient from falsely indicating a failed LRU or bus.

The OAS retry scheme is software selectable. A timer determines when a message is not received (the status word has not been received within a specified length of time), and a retry is attempted on the same bus. An interrupt is generated after the third failure. This can be inhibited by software, so that an interrupt is sent to the CPU after the first failure. The latter approach is being implemented in software with one retry being made on the alternate bus and a maximum of six retries allowed per computer frame. This sequence is "fail once, retry on alternate; fail twice, go to next command; if the maximum of six retries per computer frame has been exceeded, don't retry and go to next command."

6.4.3.2 Reconfiguration

The software is configured during normal full-up operation with two active AP's sharing the total software functional responsibility of the OAS computational subsystem. If an active AP fails when operating in the full-up mode, the software will reconfigure the remaining AP into the backup mode of operation to provide execution of all time-critical functions within 500 ms and all critical functions within a specified time after the AP failure.

To support the reconfiguration requirement, the software functional elements will be defined as consisting of one or more of the following processing types:

a. Time Critical. Time-critical functions will be resident in both AP's and are therefore available for execution in the backup mode immediately following reconfiguration.

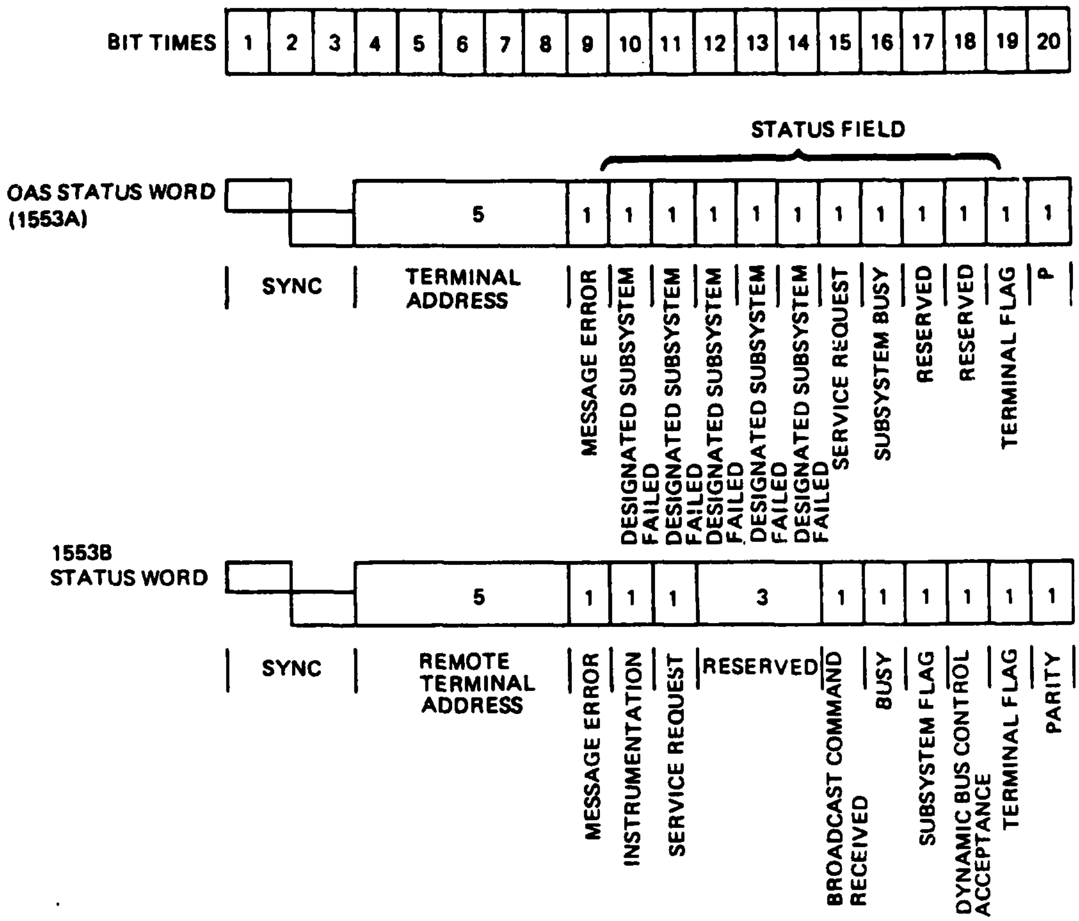


Figure 6.4-6. OAS 1553A and 1553B Status Word Comparison

b. Critical. Critical functions are not required to be resident in both AP's but will be available for execution in the backup mode within a specified time following reconfiguration.

c. Noncritical. Noncritical functions are not required to be resident in both AP's and, furthermore, are not required to be available for execution in the backup mode of operation.

When the OCP is operating in the backup mode, the time-critical and critical functions will be executed at their normal frequencies.

6.4.3.3 Built-In-Test

The BIT function handles monitoring of status data and provides data for fault message displays and fault recording.

BIT Requirements Summary

The BIT function provides --

a. AP self-test
b. Status monitoring of new prime mission equipment
c. Data for fault message displays
d. Data for recording fault status

6.4.3.4 Ground Maintenance Computer Program

The ground maintenance computer program (GMCP) is a separate load module from the flight OCP. It is used in the process of fault detection and fault isolation of new OAS PME. The GMCP checks the AP's and the 1553 data bus and assists checking the OAS equipment capable of communicating status and data over the bus.

GMCP Requirements Summary

The GMCP will provide --

a. AP self-test
b. Status monitoring of equipment containing built-in-test equipment capable of being monitored by software
c. Monitoring of equipment exercised by software
d. Interface for operator control of program execution
e. Interface for status displays to operators
f. Retrieve fault data recorded on GMCP tape in flight

6.4.3.5 Operational Computer Program Software Overview

The five major functions of the OCP software are executive, navigation, weapon delivery, control and displays and built-in-test (see fig. 6.4-7).

These functions are handled by separate programs that are mostly independent, except that the executive handles all I/O and therefore will interface to each functional program. These programs are scheduled to execute during specific periods within a major frame. Each processor has schedules for priority I/O, normal I/O, interrupt processing, foreground processing, and background processing as described in the following paragraphs.

I/O Processing

a. Priority I/O. When priority I/O is initiated, only normal I/O in progress is suspended and then resumed when priority I/O has completed. Priority I/O is used during MIU I/O and inertial navigation unit (SPN-GEANS) "torquing."

b. Normal I/O. These I/O operations may be time scheduled at specific times during the minor frames, or the operation can be done any time that the controller is not busy.

CPU Processing

a. Interrupt Processing. The real-time clock interrupts at 64 Hz, which initiates the executive routine to start each minor frame. The executive sets up the I/O and transfers control to the various foreground and background processing tasks.

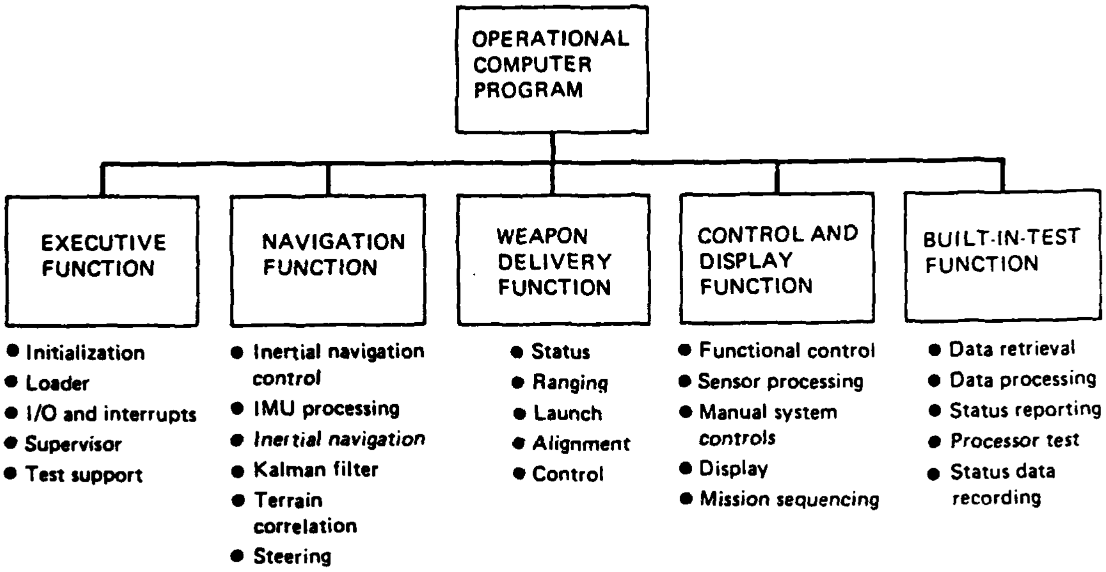


Figure 6.4-7. Operational Computer Program

b. Foreground Processing. Foreground processing may be interrupted by any source and is executed regularly at specific times during the minor frame. Navigation, weapon delivery, control and displays, and BIT are done in foreground.

c. Background Processing. This processing has a lower priority than foreground processing. Background receives control asynchronously and may extend over several minor frames. Terrain correlation, weapons impact point calculations, Kalman filtering, and self-test are performed in background.

In the backup mode, the operational AP contains both NAWD and CAD functions. BIT and inter-AP I/O have been eliminated. Less time is available for background operations, so the background self-test will take longer than in the full-up mode.

6.4.3.6 Executive Function Description

The B-52 OAS flight software is controlled by the executive function (see fig. 6.4-8). The executive is driven by a real-time clock interrupt occurring every minor frame, with four minor frames constituting a major frame. Upon occurrence of a minor frame interrupt, the supervisor module of the executive function is executed.

Within this framework, the supervisor module performs all functions that are characterized by precise timing specification. It passes control to the I/O module to initiate all I/O activity accomplished on a cyclic basis. These activities include updating AP main memory via serial I/O with inputs from avionic subsystems and transmitting data via serial I/O on a periodic basis to all avionic subsystems. Upon completion of all cyclic functions, the executive, via the supervisor module, initiates background tasks. Background

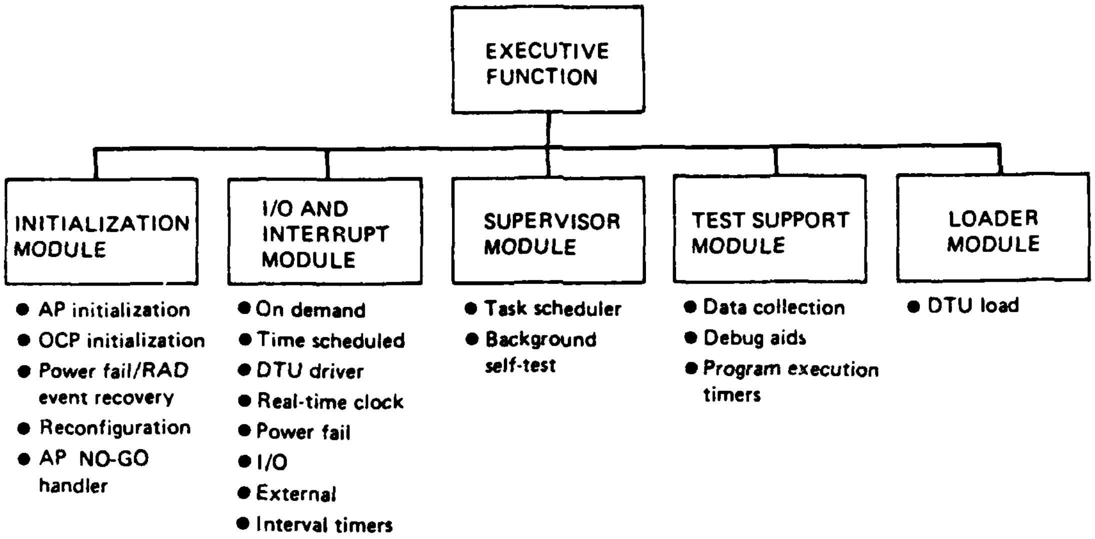


Figure 6.4-8. Executive Function Modules

tasks do not have to be completed in any one minor frame, but may have a much larger time span (e.g., Kalman background processing that has a cyclic time frame of 6 seconds).

In addition, the executive function provides all initialization functions including loading, validation, and synchronization of both AP's. Further, the executive function provides the capability, via unscheduled event recovery, to perform a software recovery upon detection of radiation events, electromagnetic pulse events, and power transients because of transfer.

The executive function also provides the capability, via the I/O and interrupt processor modules, to process serial I/O errors by retrying the message.

6.4.3.7 Serial Input-Output Processing

The serial I/O processor (1553A bus controller) controls communication between the AP and all external systems (including other AP's). Control is passed to the serial I/O module of the task scheduler at the beginning of every minor frame before the major functions are scheduled.

Serial I/O Overview

The serial I/O processor first calls the DTU I/O handler so that it may set up ondemand I/O requests to support DTU I/O. Secondary portions of the chains needed during this minor frame are linked together to one of the three chains per bus pair. These chains are--

1. Bus-independent I/O chain
2. Bus-dependent I/O chain for primary bus
3. Bus-dependent I/O chain for the alternate bus

Each type of I/O chain is in both the NAWD bus controller and the CAD bus controller.

Finally, I/O is initiated. The bus-independent chain starts with the time-scheduled I/O and terminates with an interrupt at the completion of bus-independent ondemand.

Upon receipt of the interrupt, the bus-dependent I/O chain for the primary bus is initiated. This I/O also terminates with an interrupt.

Upon receipt of this second interrupt, the bus-dependent I/O for the alternate chain is initiated. When the completion interrupt for this chain is received, all I/O, with the possible exception of priority I/O, is finished for the current minor frame.

Priority I/O

Priority I/O is a feature of the AP-101C processor that enables a normal I/O chain to be suspended at the completion of a message and allows the priority I/O chain to execute. At completion of the priority chain, the normal chain can resume execution.

The navigation software has a special need for fast-turnaround I/O to the SPN-GEANS inertial measurement system. Input from the SPN-GEANS is performed via normal I/O. As soon as input is complete, navigation processing occurs; upon completion of processing, priority I/O is used to output data to the SPN-GEANS.

An interval timer interrupt is used to schedule the priority I/O to input data from the MIU. An I/O complete interrupt at the completion of the input initiates weapon delivery verification of these data. When verified, priority I/O is used to instruct the MIU to accept these data.

I/O Chain Instruction Format

For each I/O operation, the status word returned from the RT is stored in main memory. If an I/O operation fails, a failure indicator is also stored. The I/O check interrupt routine will determine where to store the failure indicator. Because it is desirable not to induce system overhead in the I/O check interrupt routine, the routine must be able to determine indicator location with a minimum amount of software overhead. This led to standardized I/O chains, as shown in figure 6.4-9.

There are two basic types of I/O:

a. Bus controller (BC) to and from RT I/O
b. RT-to-RT I/O

BC to and from RT I/O Standardized Chain Structure. This type of I/O (BC-to-RT or RT-to-BC I/O) returns one status word from the RT to the BC.

The I/O chain command to perform the I/O is two full words in length. It is immediately followed by a one full word I/O chain command to store a 16-bit 1553A status word in a given memory location. If the I/O fails, the I/O check interrupt routine determines the address of the chain instruction that

failed. By an offset to this address, the status word save location is calculated. The failure flag is stored in the status word save location +1. Figure 6.4-9 depicts the layout of this structure.

RT-to-RT I/O Standardized Chain Structure. This type of I/O (RT-to-RT) returns two status words, one from each RT to the BC. The RT that transmits these data sends its status word first, followed by the receiving terminal, which sends its status word later. The I/O chain command to do the I/O is two full words in length. It is immediately followed by two full word commands: the first to save the transmitter status, followed by the command to save the receiver status.

If the I/O fails, the failure indicator is saved at the transmitter status word save location +1. Figure 6.4-9 depicts the layout of this structure.

Time-Scheduled I/O

Time-scheduled (T/S) I/O is subdivided into two categories:

a. Weapon delivery I/O
b. General T/S I/O

Weapon Delivery I/O. Weapon delivery I/O, if any, starts the I/O in every minor frame. The only exception is that the SPN-GEANS gimbal torquing data input is done first for every other minor frame in the NAWD.

The weapon delivery I/O processor will generate the weapon delivery chains and include the SPN-GEANS gimbal torquing data input if necessary.

① BC-to-RT or RT-to-BC I/O

<table>
<tr><td>1</td><td>000</td><td>SMSG OPCODE</td><td>0</td><td>INTPT
P N</td><td>0</td><td>EXTENDED
DATA ADDR</td></tr>
<tr><td colspan="7">COMMAND</td></tr>
<tr><td colspan="7">DATA BUFFER ADDRESS</td></tr>
<tr><td colspan="7">RESERVED</td></tr>
<tr><td>1</td><td></td><td>READ REG
OPCODE</td><td colspan="2">EXTEND ADDR</td><td colspan="2">0 1 1 1</td></tr>
<tr><td colspan="7">ADDR TO SAVE 16-BIT STATUS WORD</td></tr>
</table>

② RT-to-RT I/O

<table>
<tr><td>1</td><td>0 0 0</td><td>RT TO RT
SMSG OPCODE</td><td>0</td><td>INTPT
P N</td><td>0</td><td>0 0 0 0</td></tr>
<tr><td colspan="7">RECEIVE COMMAND</td></tr>
<tr><td colspan="7">TRANSMIT COMMAND</td></tr>
<tr><td colspan="7">RESERVED</td></tr>
<tr><td>1</td><td></td><td>READ REG
OPCODE</td><td colspan="2">EXTEND ADDR</td><td colspan="2">0 1 1 1</td></tr>
<tr><td colspan="7">ADDR TO SAVE 16-BIT STATUS WORD 1</td></tr>
<tr><td>1</td><td></td><td>READ REG
OPCODE</td><td colspan="2">EXTEND ADDR</td><td colspan="2">1 1 1 0</td></tr>
<tr><td colspan="7">ADDR TO SAVE 16-BIT STATUS WORD 2</td></tr>
</table>

Figure 6.4-9. Standardized Chain Structures

General T/S I/O. The general T/S I/O follows the weapon delivery I/O (if any) each minor frame.

This type of I/O is characterized by remaining constant over each minor frame. The I/O chain instructions are constructed at assembly time.

Ondemand I/O. The ondemand I/O emcompasses two types of I/O:

a. Priority ondemand I/O
b. Standard ondemand I/O

Priority Ondemand I/O. This type of I/O is time-critical. Upon making a priority ondemand I/O request, the currently executing I/O chain, if any, is suspended and the priority I/O chain is initiated. When the priority chain terminates, the suspended chain, if any, resumes.

Example: To improve circular error probability, the I/O for the SPN-GEANS gimbal torquing data must have fast turnaround. Data are input from the SPN-GEANS, NAV processing occurs, and a priority ondemand I/O request is initiated to output the gimbal torquing data to the SPN-GEANS.

Standard Ondemand I/O. This type of I/O is not time-critical and may occur anywhere in a given minor frame. This type of I/O is characterized by the fact that it does not occur every major frame and that users requiring the I/O must issue a request for the I/O to occur.

Inter-AP Communications

When a channel is commanded to go to remote mode, an address must be given for a data address table. The data address table is an array of 64 pointers to data blocks. These data blocks may be read or written by the AP that is in bus controller mode on the bus.

Inter-AP I/O Mechanization

Figure 6.4-10 shows the relation between the data address table, a data block, and the command word transmitted by the bus controller. The T/R bit with the subaddress provides an index into the data address table.

The first word of the receive portion of the data address table points to full word 16 of the data address table. The second word of the data address table points to full word 16 of the transmit address table. These pointers are set up so the AP in bus controller mode can access the data address table of the AP in remote mode and can therefore access any data in core.

6.4.3.8 Control of Data Transfer Unit (DTU)

A unique mode of system control in the OAS is control of the DTU. The DTU is a nine-track magnetic tape unit consisting of a plug-in cartridge (which contains the tape, read and write heads, and tape transport mechanism), a cartridge mount, and the control unit (which contains the power supply, 1553A data bus interfaces, and the control electronics). Storage capacity is at least 1,300,000 16-bit words with a physical record size of 1,024 words.

There are 18 commands that can be given to the DTU over the 1553A bus.

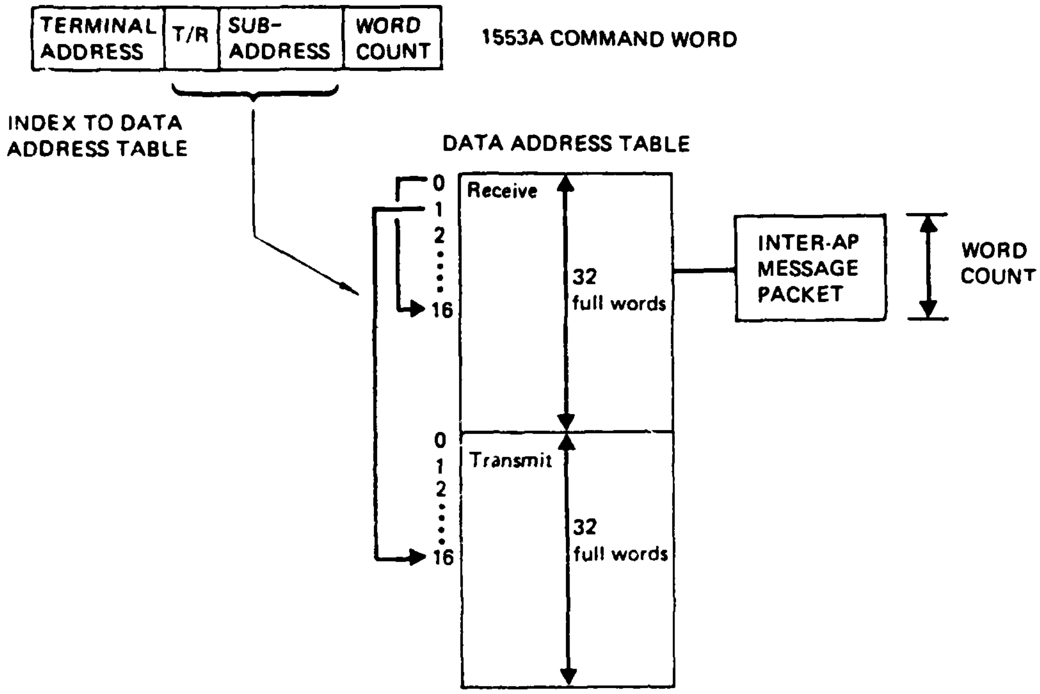


Figure 6.4-10. Inter-AP Communication Format

The substatus must be polled by the AP to determine when an operation has been completed. This is the responsibility of the DTU handler. The DTU control electronics contains a 4,096 word buffer that is used as intermediate storage between the bus interface and the physical records on the tape. To read a physical record, a read command is given and a number of transfer commands are given corresponding to the number of logical records desired. The DTU allows complete flexibility in reading or writing logical records to and from the 4,096-word buffer; for example, the user can position the buffer pointer to any word within the DTU buffer except the last 32 words.

Data Transfer Unit I/O

DTU I/O has a complexity that greatly exceeds ondemand I/O. To read from or write data to the tape, the DTU must be set up to do so. A DTU I/O handler is provided by the executive so that a user program will not need to be concerned with the mechanics of DTU I/O. All user requests for DTU I/O will be via the DTU I/O handler.

The CAD processor is the bus controller on the DTU bus. Therefore, the NAWD processor is incapable of directly accessing the DTU. If a user in the NAWD desires DTU I/O, the NAWD DTU I/O handler communicates the DTU I/O requests to the DTU I/O handler in the CAD processor.

If the processors are configured into the backup mode, the active processor is a bus controller on the DTU bus.

NAWD DTU I/O Mechanization. For the NAWD AP to use the DTU in a normally configured mode, the following scheme is used:

a. A NAWD user requests a DTU I/O setup via the NAWD DTU I/O handler.

b. The NAWD DTU I/O handler sends a setup message to the CAD DTU I/O handler via inter-AP I/O.

c. The CAD DTU I/O handler sets up the DTU I/O by initiating the process to position the requested tape to the proper place.

d. The CAD DTU I/O handler notes that the requested tape is properly postioned.

e. The CAD DTU I/O handler sends a setup complete message to the NAWD DTU I/O handler.

f. The NAWD DTU I/O handler sets a flag indicating to the NAWD user that setup is complete.

g. The NAWD user notices that the previous request is complete and initiates a transfer request via the NAWD DTU I/O handler.

h. The NAWD DTU I/O handler sends a transfer message to the CAD DTU I/O handler.

i. The CAD DTU I/O handler sets up the requested DTU I/O and message status I/O to the NAWD.

j. The NAWD DTU I/O handler sees the message status I/O and sets a message transferred flag for the NAWD user that the I/O is complete.

k. The NAWD user sees the transfer complete flag and either initiates another transfer request (g) or issues a notification to the NAWD DTU I/O handler that the I/O transaction is complete.

l. The NAWD DTU I/O handler sends a transaction complete flag to the CAD DTU I/O handler.

m. The CAD DTU I/O handler makes the tape available for another user.

Figure 6.4-11 shows the relative placement of the these events.

DTU I/O Handler. As seen by a user, the DTU I/O handler is no different in procedures or parameters for either processor in the normal or backup state.

The DTU I/O handler will periodically check the status of the DTU's to determine if a cartridge has been removed or inserted. If a cartridge has been removed, the DTU will be marked empty. If a cartridge has been inserted, it will be rewound and identified.

The DTU I/O handler will queue up requests for each tape unit on a first-come-first-served basis. One user will be allowed to terminate its operations before the next one begins. The physical record size must be an integral number of logical records.

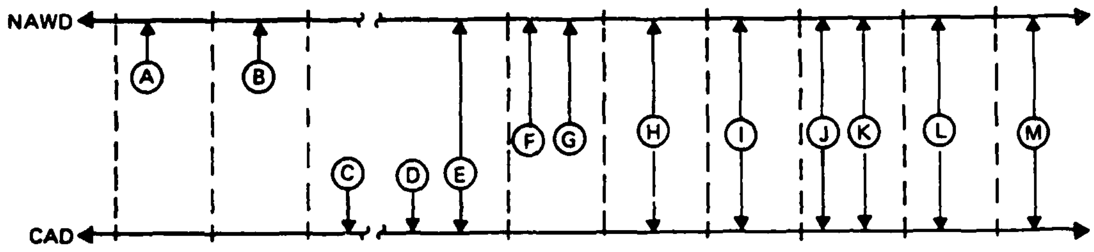


Figure 6.4-11. NAWD DTU I/O Mechanization

Reading from the DTU's. To read a file from a DTU tape, the following is done:

a. A user issues an open file request that specifies the tape identification code, the file identification code, and the desired initial record number.

b. When the open file request is complete, the user requests to move the data between the AP and the DTU.

c. When all data have been moved, the user issues a close file request that frees the tape for other requests.

Writing to the DTU. The tape that accepts write commands has two files on it. The first is the EXEC tape ID file that identifies the tape. The second is the write file onto which all write requests are written.

Data may be written on the R/W DTU tape in physical records of 32 to 4096 words. Physical record size must be an integral number of 32-word logical records and will be selected by each user. The user must have a buffer the size of the physical record to ensure data integrity after unscheduled events.

A flag is used to indicate the status of a write request to the user. This flag will indicate whether the I/O request completed, failed, or is still in progress.

6.4.4 Bus Controller

The AP-101C processor unit is a general processor with floating point capability, 65,536 32-bit words of core memory, and six 1553A bus controllers for communication with other subsystem elements. The instruction set is similar to the IBM 360 series of computers. Several special-purpose instructions have been provided, including square roots, sine, cosine, and exponents. The AP-101C contains two sets of eight general registers of 32 bits and one set of eight registers of 32 bits for floating point operations. A 64-bit program status word (PSW) contains the next

instruction address; condition code; interrupt masks; the interrupt code specifying what caused the interrupt; and bits to specify the register set, problem and supervisor status, and the wait state.

The word length is 32 bits; instruction length may be 16 or 32 bits with addressing by half words of 16 bits. Sixteen bits will address 32K full words, and up to 256K full words may be accessed by using an extra 4 bits located in the PSW. These 4 bits are obtained from the data sector register for instructions involving data access or from the branch sector register for branch instructions. If the most significant bit of a 16-bit address is 0, none of the sector registers are used and the effective address is the lowest 16K full word of core. Each 16K full word constitutes one data sector so that the computer will have four sectors for 64K of core. Each sector will contain specific program functions as follows:

Sector	Functions
0	Executive/common
1	Navigation
2	Weapon delivery
3	CAD/BIT

Interrupts

A total of 34 individual sources of interrupts with 23 double words of memory are specifically set aside to store the PSW that is loaded when the interrupt occurs. To completely identify the source of the interrupt, the interrupt code bits of the PSW must be used. There are 19 levels of interrupt priority. The 16 mask bits in the PSW allow the user to mask some of the interrupt sources; 17 interrupt sources are unmaskable.

Interrupts cause the current PSW to be saved in an area of core called the preferred storage area (PSA) and a new PSW is picked up from an adjacent location in the PSA. When the interrupt service subroutine has been completed, the PSW is moved from core to the PSW register with a load program status word instruction thereby returning to the interrupted program.

Figure 6.4-12 shows a simplified OAS hardware structure emphasizing the processors and I/O structure. Each AP has six bus controllers: two are unused, one is in controller mode, two are in remote mode, and one is in quiescent mode. Data transfers can be initiated only in the controller mode.

One bus pair is connected to the NAWD interface units; this bus pair is called the NAWD bus pair. One bus pair is connected to the CAD interface units and the four DTU's; this bus pair is called the CAD bus pair. Each bus controller is attached to a 1553A serial data bus: one bus of a pair is primary, the other bus is an alternate in case of a hardware failure on the primary bus.

Each processor has separate loads that are identical for time-critical functions. One AP executes the NAWD programs, and one AP executes the CAD programs. In full-up mode, the NAWD AP controls the buses connected to the NAWD interface units, and the CAD AP controls the buses connected to the CAD interface units and the DTU's. In backup mode, the operational AP controls both the NAWD bus pair and the CAD bus pair. Each AP has 16 discrete output bits, 16 discrete input bits, and 7 interrupt bits.

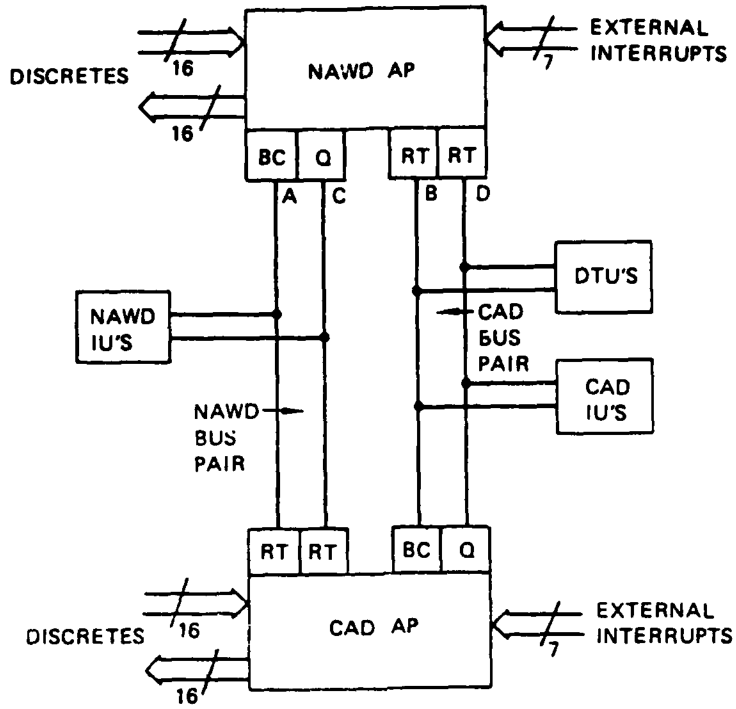


Figure 6.4-12. OAS Hardware Structure

In the remote mode, the channel responds to commands received over the serial bus. In the quiescent mode, the channel responds to commands from the CPU and ignores any traffic over the 1553 serial bus.

<u>Serial I/O</u>

Serial I/O for a specific controller (also called a channel) is initiated by--

a. Storing in memory a chain instruction with a pointer to a data buffer; a bit in the chain instruction specifies whether a completion interrupt will be allowed

b. Storing a command cell instruction at a specific location in memory corresponding to the channel. The second word of the command cell points to a chain instruction

c. Executing an internal control instruction giving a specific channel number

Three kinds of serial I/O chains may be identified, depending on priority and synchronization of the request with the clock:

a. Time Scheduled. This nonpriority I/O is predetermined and prescheduled for each minor frame by the programmer. This I/O is initiated on a 64-Hz basis. Upon completion of the time-scheduled chain for a specific frame, the ondemand chain is initiated.

b. Ondemand. This I/O is initiated by an asynchronous request at run time. This I/O is nonpriority and therefore does not interfere with execution of the time-scheduled chain. Ondemand I/O is not scheduled during the current frame but at a specific future minor frame.

c. Priority. This I/O is initiated asynchronously by the program and causes suspension of the current nonpriority chain (after the current channel instruction has been completed). The nonpriority chain is resumed when the priority chain has been completed.

6.4.5 Remote Terminal

The OAS RT's are of five different types, made by five different manufacturers. Four of these types of RT's are integral to the subsystems. The fifth type of RT can be integral or standalone, depending on the application.

The first type of RT is integral to the two AP's. The RT function is a processor-controlled mode of the I/O channels of the IBM AP-101C processor. This I/O structure was discussed in section 6.4.4.

The second, third, and fourth types of RT's are also integral to their subsystems. The second type is in the common Doppler, called the DVS in the OAS. The third type of integral RT is in the four DTU's that are built by Sunstrand. A top-level description of DTU operation was presented in section 6.4.3.8. The fourth type is in the DEU and is procured from Sperry.

The fifth type of RT is built by Boeing and is in nine of the units connected to the OAS data buses. The nine units are the RIU, the EIU, the AIU, the CDIU, two IEU's, and three MIU's. These units have unique subsystem interfaces tailored to a particular application; however, all 10 employ a common interface to the 1553A buses called a common core.

The two-card B-52 OAS RT consists of a dual modem card to interface with two multiplex data buses and a handshaker card containing a 256-byte buffer memory. Except for initialization, the OAS RT operates independent of the user who interfaces with the handshaker card. Data words received or transmitted over the data bus are stored in or obtained from the buffer memory.

The basic architecture of the OAS RT is shown in figure 6.4-13. The output of the two independent modems is combined off the card and passed to the handshaker. When the modem receives a command word with its terminal address, it will signal the handshaker that there has been an address compare on that channel. The handshaker has control over the output enables of both modems and will only listen to the channel with the most recent address compare. In the receive mode, the serial data from the multiplex bus are shifted into shift registers. It is read out a byte at a time by the handshaker. In the transmit mode, the handshaker loads the shift register in parallel and the modem transmits it serially.

The modem detects and generates syncs, decodes the terminal address, counts bits within a word, checks for valid Manchester data, and detects and generates parity. The handshaker decodes the T/R bit, subaddress, and word count from the command word and transfers the data words between its buffer

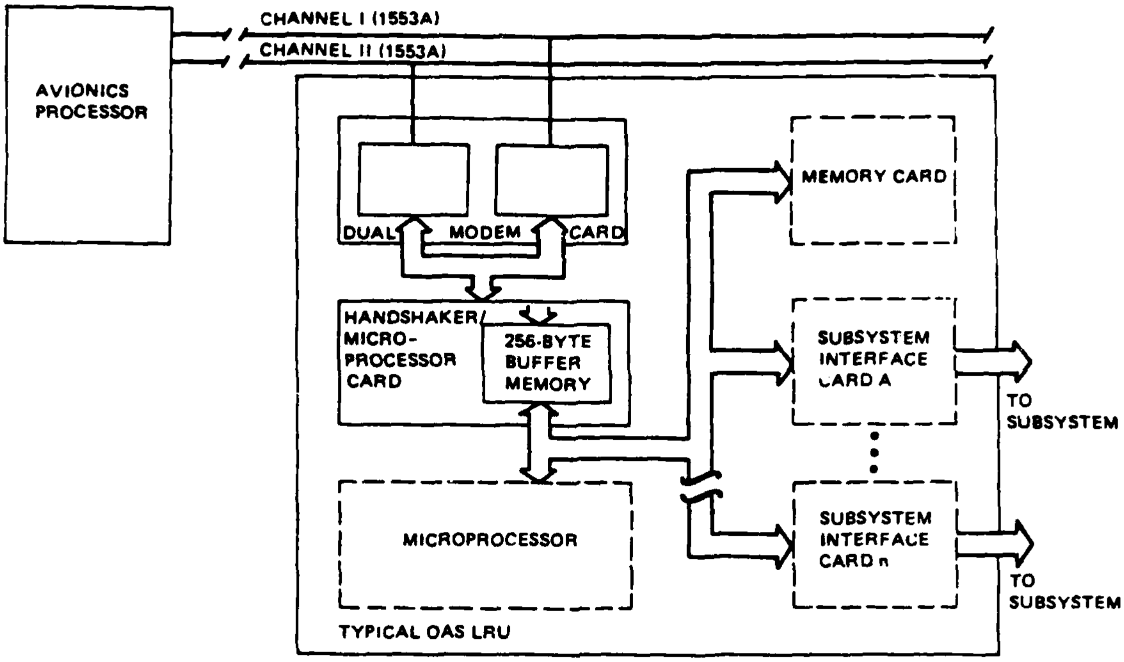


Figure 6.4-13. OAS Remote Terminal Block Diagram

memory and the modem card. During a modem transfer, the handshaker controls the internal bus and passes the data words from and to the modem to and from memory at a location depending on the address generated by the mapping information derived from the given subaddress and T/R bit. Data are transferred over the internal bus a byte at a time, starting with the least significant byte of data.

In addition to the data words in RAM (i.e., the buffer memory), the user can read the most recent command word.

The handshaker is the interface between the modem and the subsystem. It contains a 256-byte buffer memory (RAM), which all modem data words are transferred to or from. The subsystem has access to the contents of the buffer memory so that data words can be read or updated. The modem and handshaker communicate through several handshake signals.

There is some initialization required by the handshaker, but normally the OAS RT (modem and handshaker) can operate independent of the subsystem. The internal bus is 8 bits wide and is used to pass data a byte at a time. It is isolated from the subsystem bus by tristate buffers. When the RT is idle (i.e., not handling a transmission on the data bus), the subsystem has immediate access to the RAM. During data bus transmissions, the handshaker will take control of the internal bus to prevent subsystem access to the RAM.

The handshaker decodes and stores the subaddress, T/R bit, and word count from the command word. It counts data words and transfers them between the modem and RAM. It also supplies the status word and handles the send status word and enable and disable alternate transmitter mode codes.

The handshaker is designed to interface with a dual-channel modem but does not interface with both channels simultanecusly. The tristate outputs of the two modem channels are tied together and controlled by the handshaker, which will enable the channel with the most recent address compare. Thus, handshaking on one channel will cease if there is an address compare on the other. Handshaker firmware is designed such that an interrupted message will terminate in a known orderly fashion.

The 256-byte RAM is used to store data words for the modem. In addition, the first 64 bytes are dedicated to mapping information. Each T/R bit and subaddress combination in a command word can cause the assocated data words to be transferred (mapped) to and from any of the 192 remaining bytes in the RAM. The location of the first byte of data is mapped into the address defined by the contents of the byte at the address made up of the T/R bits and subaddress. For example, assume a command word had a T/R bit = 1 and a subaddress = 00101, then the contents of 25H (00100101) contains the address of the first byte to be transmitted by the modem. If the contents, of 25H is B4 and the contents of B4 and B5 are 58 and 7C, then a command word in the example will cause a data word of 7C58 to be transmitted. Note that the low byte of data word is stored in the lower address.

6.5 DIGITAL AVIONIC INFORMATION SYSTEM MULTIPLEX SYSTEM -EXAMPLE 5

6.5.1 Application Area

The digital avionic information system (DAIS) is a general-purpose *information transfer system* oriented toward synchronous data processing. DAIS was developed by the Air Force Avionics Laboratory for use primarily in the areas of avionics and flight controls. This description of DAIS is partially derived from the paper "Digital Avionic Information System (DAIS) Multiplex System," by Captain Frederick L. Pensworth of the Air Force Avionics Laboratory, WPAFB, published in the AFSC Multiplex Data Bus Conference Proceedings, November 1976. The version of DAIS described here is that presented in the referenced paper.

Because of the generality built into the DAIS software structure, the system could be oriented toward any synchronous multiplexed application with a repetition rate of less than 128 times per second and a one- to four-processor federated processing requirement; however, DAIS has not been used outside of the laboratory environment and none of the equipment is MIL-qualified.

6.5.2 System Architecture

DAIS is an avionic system architecture that consists of multiple federated processors that communicate between each processor and the other system elements (sensors, weapons, and controls and displays) through a standardized multiplex data bus system. This system architecture is flexible enough to accommodate a wide variety of avionic configurations, missions, and sensors, and provides redundancy to improve avionic information availability. This flexibility is achieved by defining functionally standardized core elements that can be integrated in various mixes and configurations to accomplish the specific mission requirements. DAIS system architecture is depicted in figure 6.5-1. The master executive provides a centralized control point for system operation; however, this control point may be relocated for system reconfiguration.

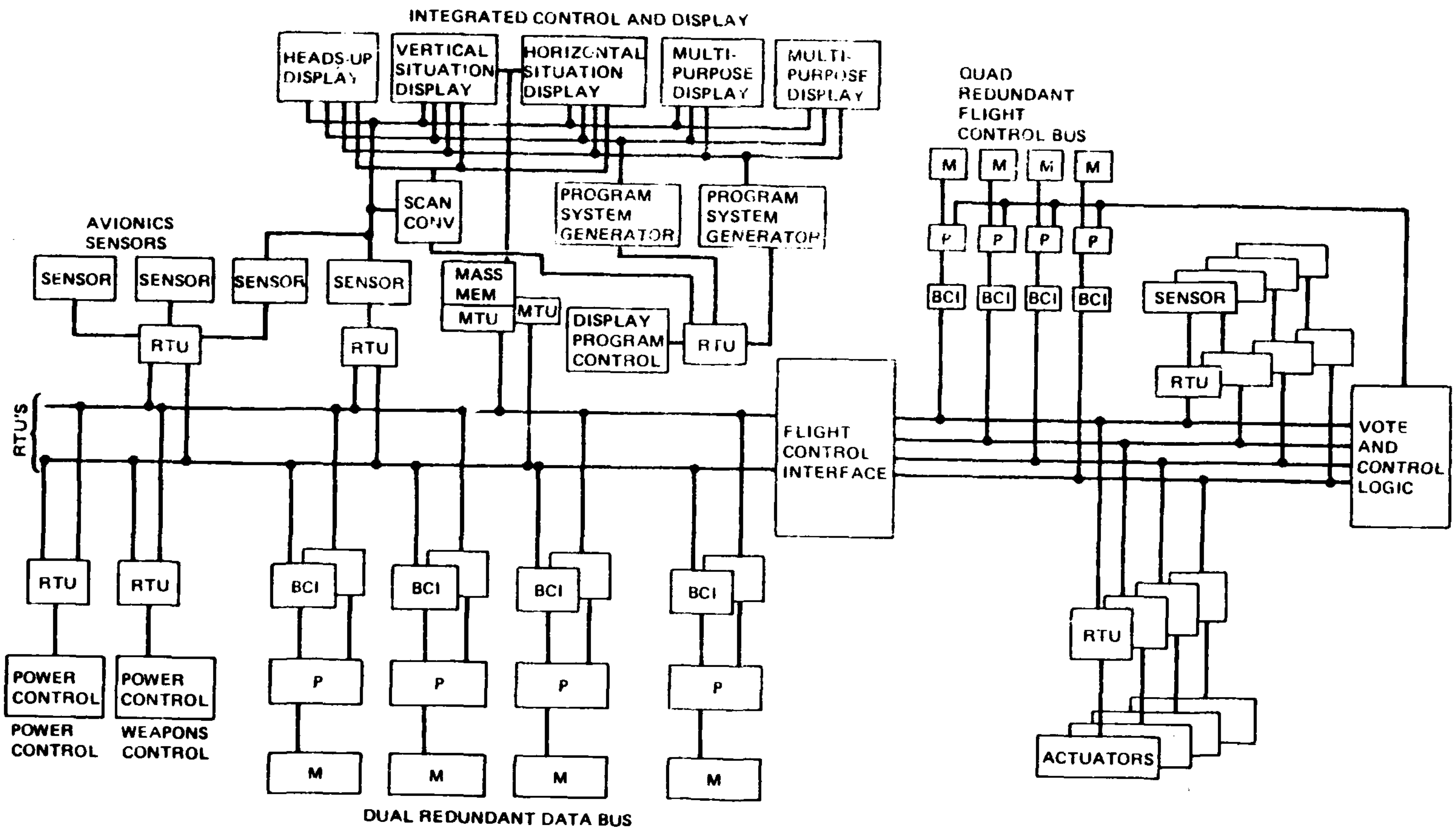


Figure 6.5-1. Digital Avionics Information System (DAIS) Architecture

The DAIS core elements can be described functionally as follows:

a. DAIS processors (Westinghouse AN/AYK-15) are general-purpose computers designed for airborne use. The number of DAIS processors required is determined by the mission requirements and the mission software to be loaded. The DAIS processors are arranged in a federated processor configuration, interconnected through the DAIS multiplex system.

b. DAIS multiplex provides command and data information transfer between the DAIS processors, the subsystems (sensors and weapons), and the controls and displays. It consists of bus control interface units (BCIU), one for each DAIS processor; RT units, which interface the subsystems to the data bus; and the redundant multiplex data buses. The system control procedures define the bus protocol and bus error and core element failure management. This system is a time-division multiplex (TDM) and command/response system with one BCIU. The BCIU, under control of the master executive, allows bus traffic on only one bus at any given time. The other bus is provided for redundancy.

c. DAIS controls and displays are an integrated set of state-of-the-art equipment. It is designed to provide flexibility to accommodate changes, such as adding redundancy where required. It provides the capability for system management by one pilot.

d. Mission software resides in the memory of the DAIS processors. Two separate computer programs have been identified to support the DAIS architecture. These are designated the operational test program (OTP) and the operational flight program (OFP).

The sensor, weapon, and communication subsystem(s) are interfaced to the DAIS processors through the RT's. The subsystem configuration is dependent on mission requirements.

DAIS generally complies with 1553A; 23 mode codes were defined in DAIS. Mode code definition for DAIS and 1553B is presented in table 6.5-1.

The DAIS status word is significantly different from the 1553B. DAIS uses a busy bit and a service request bit in the same way as 1553B; however, 1553B does not allow a 5-bit vector field as does DAIS. Status word definition for DAIS and 1553B is presented in table 6.5-2.

Up to 32 data words are transmitted in a message in both DAIS and 1553B; however, DAIS allows a special exception in which the first data word is replaced by the last command register after a mode code 11 (interrogate serial I/O activity register).

The electrical characteristics also differ between DAIS and 1553B. DAIS and 1553B electrical characteristics are compared in table 6.5-3.

Redundancy is provided at several levels in DAIS. The avionic bus is dual redundant. Redundant bus controller software resides in a second DAIS processor. Separate bus interface hardware is connected to each bus, but only a single bus controller module interfaces with both bus interface modules. Similar partial hardware redundancy occurs in the RT, as shown in figure 6.5-2.

Table 6.5-1. Mode Code Definition

DAIS		1553B	
Mode code number	BCIU (master mode)	Mode code number	Function
0	Valid (status response only)	0	Dynamic bus control
1	Transmit status word	1	Synchronize
2	Reset status code field	2	Transmit status word
3	Transmit BIT word	3	Initiate self-test
4	Remove power MTU 1	4	Transmitter shutdown
5	Remove power MTU 2	5	Override transmitter shutdown
6	Shutdown override MTU 1	6	Inhibit T/F flag
7	Shutdown override MTU 2	7	Override inhibit T/F flag
8	Initiate terminal self-test	8	Reset remote terminal
9	Initialize terminal	9	Reserved
10	Transmit last command	10	Reserved
11	Interrogate activity register	11	Reserved
12	Reset serial input channel	12	Reserved
13	Interrogate module error register	13	Reserved
14	Initiate serial channel I/O	14	Reserved
15	BIT masking	15	Reserved
16	Word masking	16	Transmit vector word
17	No-op	17	Synchronize
18	Master function interrupt	18	Transmit last command
19	Valid (status response only)	19	Transmit BIT word
20	Busy override	20	Selected transmitter shutdown
21	System interrupt	21	Override selected transmitter shutdown
22	BIT word reset	22	Reserved
23-31	Spare	23-31	Reserved

Table 6.5-2. Status Word Definition

DAIS

- Unique RT status codes

Bit:	Status:
7	1 = channel activity
8	1 = channel parity error
9	Message error

- Unique remote BCIU status codes

Bit:	Status:
7-16	All 1's = busy mode
17	All 0's = no condition
18	Other codes: Asychronous request vector
19	Terminal failure

- Unique station logic unit (SLU) status codes

Bit:	Status:
13	1 = BIT transfer error (BT)
14	1 = SLU bus shutdown (BC)
15	1 = power cycle (PC)

- Unique mass memory codes

Bit:	Status:
14	Invalid command (IC)
15	Not ready (NR)

1553B

Bit:	Status:
9	Message error
10	Instrumentation
11	Service request
12	Reserved
13	Reserved
14	Reserved
15	Broadcast command received
16	Busy
17	Subsystem flag
18	Dynamic bus control acceptance
19	Terminal flag

The DAIS flight controls architecture is similar to the avionic architecture, except that it is quad redundant and operates using voting control logic, as shown in figure 6.5-1.

Synchronization of the multiplex system occurs via a set of messages, transmitted over the data bus, directed to each processor announcing the next minor cycle. The master processor is the only processor responsible for timekeeping. Processing of application software is minor cycle dependent. A particular application program will only be run in specified minor cycles. Minor cycle synchronization will be discussed further in section 6.5.3.1.1.

6.5.3 System Control

The system control procedures define the total DAIS operation and consist of the following system operational modes:

a. System Startup and restart operation
b. Pre-flight and post-flight test operations
c. Normal system operations
d. System backup and recovery operation
e. System reconfiguration operation

Table 6.5-3. Electrical Characteristics

	DAIS	1553B
Message format	Maximum message = 32 words Response time = 2 to 5 μs* Transmit timeout timer = 660 μs Bit error rate = 10^{-12} Message error rate = 10^{-6} *Later extended to 10 μs	Maximum message = 32 words Response time = 4 to 12 μs (same as 2 to 10 μs) Transmit timeout timer = 800 μs Minimum no response time out timer = 14 μs Minimum message gap = 4 μs Bit error rate = 10^{-12} Word error rate = 10^{-7}
Twisted pair bus	30 pF/ft Shielding = 80% Impedance = 63 to 77 ohms Attentuation = 1.0 dB/100 ft	30 pF/ft Shielding = 75% Impedance = 70 to 85 ohms Attenuation = 1.5 dB/100 ft
Waveforms	t_r, t_f = 100 ns Output level ±6V to 20V p-p Input level ±1V to 20V p-p	t_r, t_f = 100 to 300 ns Output ±18V to 27V p-p (transformer coupled) Input ±0.8V to 14V p-p
Data rate	1 MHz ±0.01%, long term, ±0.001% short term	1 MHz ±0.1%, long term; ±0.01% short term
Stubbing	2,000 ohms at 0.1 to 1 MHz Coupler boxes External fault resistors	1,000 ohms at 0.075 to 1 MHz Coupler boxes External fault resistors
Bus output noise	10 mV, p-p	14.0 mV, rms (transformer coupled) 5.0 mV, rms (direct coupled)
Common mode rejection	Dc to 2 MHz, 10V peak	Dc to 2 MHz, ±10V peak

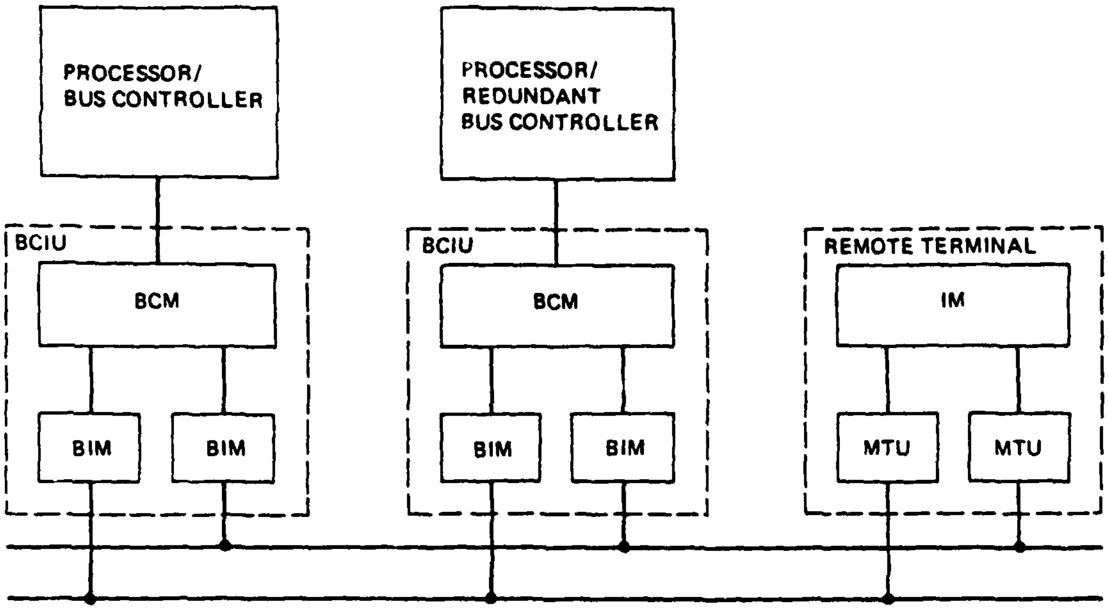


Figure 6.5-2. Redundancy in DAIS

6.5.3.1 Normal System Operations

This section defines normal system operations. During normal system operation, the master BCIU and the master processor operate together as the bus controller. The following functions which are performed during normal system operations:

a. Minor cycle synchronization

b. Bus message operation (bus protocol): synchronous, asynchronous, and mode bus message operation

c. Mission application tasks

d. System error and failure management

e. Monitor management function

f. Configuration management

g. Power management

Some of these operations are discussed in the following sections.

6.5.3.1.1 Minor Cycle Synchronization

Minor cycle synchronization is required to provide a time reference for the processors.

The master executive performs minor cycle synchronization when the minor cycle clock expires by accomplishing the following:

a. Checks synchronous bus messages for completion.

b. Increments minor cycle count and transmits minor cycle mode commands to remote and monitor processors.

c. If synchronous bus messages were not completed, it extends period for one more minor cycle time period and completes synchronous bus messages.

d. If a minor cycle synchronization failure is indicated, it tabulates failure and attempts to complete all minor cycle commands. Minor cycle synchronization failure analysis is performed after all minor cycle commands have been attempted.

e. Upon successfully completing minor cycle synchronization and setting up the synchronous bus message table, synchronous bus message transmission is initiated. The local executive checks for correct minor cycle number and sets up the synchronous bus message tables for the next minor cycle.

Minor cycle synchronization is mechanized in the DAIS system through the use of the master function interrupt mode command. The remote processor interprets the mode data as the minor cycle number.

6.5.3.1.2 Bus Message Operation

Bus traffic is composed of both synchronous, asynchronous, and mode command transmissions. The synchronous bus traffic is controlled by a predefined bus command list. Bus traffic may occur between the bus controller and a terminal or between two terminals. Bus messages are made up of two or more words. Word types are command, status, and data. The bus controller generates message sequences that exercise system control and transfer data and status information between the processors and terminals as required.

The synchronous bus message operations define --

a. Initialization and operation of BCIU for message transmission operations
b. Responses to BCIU interrupt conditions at master, monitor, or remote processor
c. Responses to status word conditions at master processor
4. Responses of RT to synchronous message

Synchronous operations are controlled by the executive master scheduler. Anomalous conditions in the synchronous bus traffic are interpreted by the BCIU and presented to the processor via dedicated priority-encoded interrupts.

Asynchronous bus operation is required for interprocessor task communication and cockpit switches (discretes) and to indicate system problems. Asynchronous bus message traffic is initiated in one of three ways:

a. An application task requests an executive service to--

1. Write a compool
2. Set an event
3. Invoke a task
4. Cancel and terminate tasks

The compool/event/task is located in other processors. To effect the required executive service, asynchronous messages(s) must be sent to the other processors.

b. An application task requests that a particular message (compool data) be transmitted to a specified terminal at a specific future time. Two asynchronous messages may be required. If the application task is not located in the master processor, one asynchronous message is required to transfer the critically timed message to the master executive and to identify the terminal to which the message is to be sent and the time at which transmission is to occur. A second asynchronous message is required to transmit the critically timed message from the master to the specified terminal.

c. An asynchronous message transmission is requested by an RT (initially by a subsystem connected to that RT); the master executive receives the request and controls the transmission of the requested message. The request is made to the master BCIU through the status word. This operation requires a status word and mode codes. It is required that the BCIU be able to interrupt the processor.

Mode command operations are initiated as a result of various requests from the master executive, such as message error analysis, power management, minor cycle synchronization, etc. The master BCIU transmits the mode command, and the addressed RT or remote BCIU responds to the mode command by performing the requested operation or providing the response (e.g., BIT word and last command word). Table 6.5-1 identifies the DAIS mode commands. Mode commands impact the master executive software, the BCIU microcode, RT microcode, and the number of registers in the BCIU and RT.

6.5.3.1.3 System Error and Failure Management

Error control is required because of errors on the bus, hardware failures, and power transients.

The master processor and BCIU perform system error and failure management by--

a. Monitoring message sequence to determine successful or unsuccessful completion

b. Repeating the messages (class I retry, as explained later), first on the same bus and then on the alternate bus if message retry is indicated

c. Performing class II retry (explained later) if required for a message

d. Analyzing detected message error, terminal failure status information, and message sequence history to detect and isolate failure to the core elements (RT, BCIU, processor, and control and display) or redundant elements of the RT, BCIU, or bus

e. Requesting self-test be performed when core element is suspected as failed and declaring core element as failed if self-test is not successful

f. Reporting declare failures to configuration management, which modifies the synchronous and asynchronous bus traffic accordingly and also notifies the application software (the configurator) for appropriate action

Error and failure management requires the use of mode commands, retries, and processor interrupts.

There are four basic error and failure types. These include status word errors, message errors, data errors and terminal failures. The first three require that one of the three types of retries be initiated. The fourth requires an interrupt to the processor.

A class I retry is used when the repeated transmission of a message will not affect the subsystem.

Class II retry is performed when repeated transmission of the same message to an RT could cause degradation in subsystem performance. The master executive will first request transmission of a mode command to obtain from the RT the last command received by that RT. If it is not the command for the last bus message transmitted, the same bus message will be retransmitted to that RT. If the RTs last command is the same as the command for the last bus message transmitted and no message errors are reported, the master executive will not retransmit the message but will continue with the scheduled bus messages.

A class III retry is used when the last message consisted of more than one transmission. If retry is indicated, the BCIU will retransmit the last message sequence.

Status word errors include no response, parity error, and format errors. If there was an error in the last message, the RT does not transmit the status word. The only exception is an invalid mode code. If a class I retry is indicated, the message will be retransmitted. If a class II or class III retry is indicated, the built-in-test will be requested to determine if there was a message, data, or terminal error.

The master BCIU and processor monitor for message errors and will perform the following if message error condition is detected:

a. Retry the message on the same bus if retry is indicated.

b. Retry the message on the alternate bus if no redundant element failure is indicated.

c. Perform class II retry if indicated.

d. Initiate redundant element or terminal failure analysis if retry of the message is not successful.

Upon receiving a message error response after an asynchronous message transmission, the master executive will--

a. Determine if the transmitter or receiver was the suspected cause of the error condition
b. Request a status word from the suspected terminal, including retry of this mode command on one side of the data bus and then on the other if required
c. Realign the asynchronous queue and retransmit the asynchronous message if the status word was successfully received from the suspected terminal

If a bus message is not successfully completed, the master executive will--

a. Determine if a master BCIU failure is indicated by transmitting a mode command to another remote BCIU
b. Initiate BCIU self-test or request that terminal self-test be performed
c. Declare the BCIU or terminal as failed if self-test is not successfully completed.

If a busy indication is received from a remote or monitor BCIU/processor, the master executive will initiate busy override operations.

The master executive, upon receiving a terminal failure indication in the status word, will request BIT word from the terminal. The master executive, upon decoding the BIT word, will perform the following:

a. Initiate self-test of the terminal if a redundant element or terminal failure is indicated.

b. If self-test is not successful, the redundant element or terminal is declared failed and configuration management is initiated.

c. Respond to element power recovery indication by (1) repeating the message, if the element is a RT or BCIU, and (2) performing power transient recovery operations, if a remote processor or monitor processor indicates that a power transient has caused loss of system synchronization.

The master executive will tabulate suspected redundant element failures and return to complete the synchronous or asynchronous message retries. The master executive will determine if the "suspect" redundant element failure is in the master BCIU or a redundant element in a terminal or remote or monitor BCIU. After exceeding a threshold of suspected failures for a redundant element, self-test will be initiated. If self-test is not successfully completed, the redundant element is declared failed and configuration management is initiated.

Periodically, self-test of each RT and BCIU will be initiated by the master executive on both redundant element (A and B) sides. Upon successfully completing self-test, any suspect redundant element failures will be cleared.

Error and failure analysis and recovery require that the BCIU interpret the status response (or lack of) for all normal bus errors and terminal failures. The BCIU must then present the results to the processor in a logical and ordered sequence. The DAIS BCIU provides dedicated

priority-encoded interrupts to signal anomalous conditions to the processor and a separate I/O interface to read hardware status.

The system error-handling and recovery procedures are not currently implemented in DAIS. The goals of the DAIS system error recovery procedures are described in the following paragraphs.

6.5.3.1.4 System Reconfiguration

The redundant bus controller unit is called the monitor processor. The monitor sets a timer to time the arrival of minor cycle updates. If an update does not arrive prior to the maximum allowable time interval, then the master processor can be considered failed.

Reconfiguration is initiated by the failure of a processor. Both the master processor and the monitor processor are tasked with the detection of processor malfunctions. The failure mode for processors is if the malfunctioning processor ceases communication with the bus. The master will detect failures of either the monitor or the remote processors, and the monitor will detect failure of the master processor and its associated BCIU. The monitor will conduct the reconfiguration if either the master or remote processors have failed. In the event of a monitor failure, the master must conduct the reconfiguration procedures. The reduced configuration must perform diagnostic tests to determine the DAIS core elements that remain operable and must continue to perform the computational and communication tasks assigned to it. At the completion of the diagnosis and upon a signal from the processor control panel to reconfigure, the processor in control will begin the reconfiguration if two or more processors are able to function.

Reconfiguration occurs after one or more processors have failed, the system is in either the recovery or backup mode, and the operator manually initiates the master executive to load a set of mission software from mass memory. The master executive performs reconfiguration in the following sequence:

a. Directs self-test of the nonmaster processors and BCIU's using the proper mode commands over the data bus. This also accomplishes the bus communication test.

b. Checks the mass memory interface.

c. Determines the configuration of mission software to be loaded.

d. Controls the transfer of mission software (master executive, local executives, and application modules) to the proper processors.

e. Performs memory checksum of loaded computer programs.

f. Provides mission data and initialization data to the application functions.

g. Initiates normal system operation using the new mission software configuration.

If reconfiguration is being accomplished by the master executive in the monitor processor, the master executive must also provide the necessary control functions to continuously accomplish mission-critical application functions in the monitor processor.

Reconfiguration of mission software allows the system to recover avionic functions that allow completion of the mission (e.g.,weapon delivery). The set of mission software to be loaded during reconfiguration is determined by the number of active processors and is a subset of the full-up processor configuration. Each configuration will be loaded into the mass memory (preflight). The application modules to be included in each configuration (one-, two-, three- or four- processor configuration) will be determined based upon mission requirements.

6.5.4 Bus Controller

Each processor that is capable of controlling the bus is a Westinghouse AN/AYK-15 with up to 64K of memory.

The BCIU built by IBM provides the interface, control, and data transfer functions required to connect a processor with two multiplexed data buses. The BCIU is directed by its interfacing processor to operate in a mode. The BCIU is capable of operating in the following modes:

a. The remote mode provides transfer of data in both directions between the processor and either of two data buses, based upon commands received from either of the data buses,and also provides status replies on the appropriate data bus in response to commands and special internal operations and interrupts to the associated processor upon receipt of certain special commands on the data buses.

b. The master mode provides control of the two data buses based upon instructions fetched from the memory of the processor through the DMA channel by the BCIU. This control mode results in the BCIU issuing bus commands to other devices on the data buses, participating in data transfers on the buses (when the instruction dictates it), checking status responses from devices on the data buses, checking formats of the data bus operations, and reporting error conditions to the processor. At any one time in any one data bus system, there is only one BCIU in the master mode.

Figure 6.5-3 illustrates the major components that comprise a BCIU. Each bus interface module (BIM) provides the interface function to one data bus. The BIM is the same as the MTU in the RT. These modules are interchangeable.

The processor interface module (PIM) provides the interface function to the processor. Changing of processor types is accommodated by redesign of only the PIM. The bus control module (BCM) provides timing, control, instruction decoding, and data transfer routing required to implement the various operation modes defined.

The BCM is the device that controls operation of the BIM's to perform data bus operations and the device that controls the PIM to communicate with the processor. The method in which the BCM performs these control functions depends upon the BCIU operational mode. Essentially, the BCM is the control device that implements the master and remote operations.

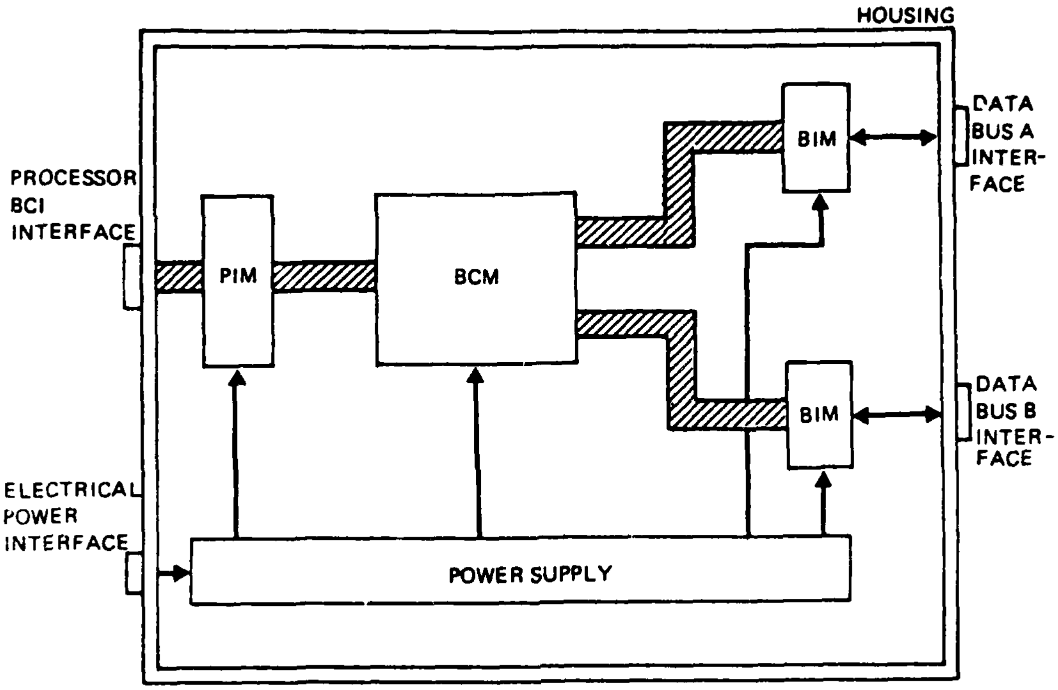


Figure 6.5-3. DAIS Bus Control Interface (BCI) Block Diagram

The BCM contains control registers that the processor can load through the programmed I/O (PIO) channel and also registers that the processor can read through the PIO channel. These PIO operations take place directly and require no action by the BCM. The PIO operations that the processor performs determine BCM state. The BCM contains the registers, selection logic, comparison logic, and control sequences required to transmit and receive words with the BCM's; read and write words through the DMA channel; provide DMA addresses for the DMA operation; and perform bus message generation. In order to have flexibility when performing these functions, a microprocessor was designed into the BCM.

The PIM forms the communication channel between the BCM and the processor. The PIM accepts control and information signals from the BCM and performs DMA operations with the processor's memory based upon these signals. The PIM also accepts interrupt request signals from the BCM and generates interrupts to the processor based upon these requests.

All DMA operations are single-word DMA read or write operations based upon a request by the BCM. When the PIM is not currently performing a DMA operation and it detects a pulse on the DMA request line, it performs either a DMA read or a DMA write operation.

The BCIU is housed in a 3/4 ATR box that is 12.5 in long. There are two BIM pages, five BCM pages, and two PIM pages.

6.5.5 Remote Terminal

The RT built by IBM provides the primary mechanism for interfacing subsystem equipment with the DAIS core elements. The RT also interfaces certain DAIS core elements with other DAIS core elements. The RT transmits and receives data on the bus and performs message validation and built-in test. The RT also receives data from the subsystems and converts it to the proper data bus format and converts data from the bus into the proper format for the subsystem.

The major components of the RT are shown in figure 6.5-4. The MTU is the device that interfaces to the data bus. The MTU transmits and receives information between the data bus and the timing and control unit (TCU). The MTU is controlled by the TCU. The unit employs common mode rejection and filtering to minimize noise susceptibility. Validity checks include sync, parity, Manchester validity, and bit count.

The TCU is the device that performs all of the timing, control, buffering, decoding, and checking required to receive information from the data bus and to reflect that information as outputs from the RT, as well as to accept inputs to the RT and reflect those inputs as information on the data bus. The TCU performs these functions based upon commands received from the data bus. In order to have the capability to add and delete mode codes and to provide greater flexibility, a microprocessor was designed into the TCU. To support mode code implementation, 16 registers were required to store such data as the last command and BIT word.

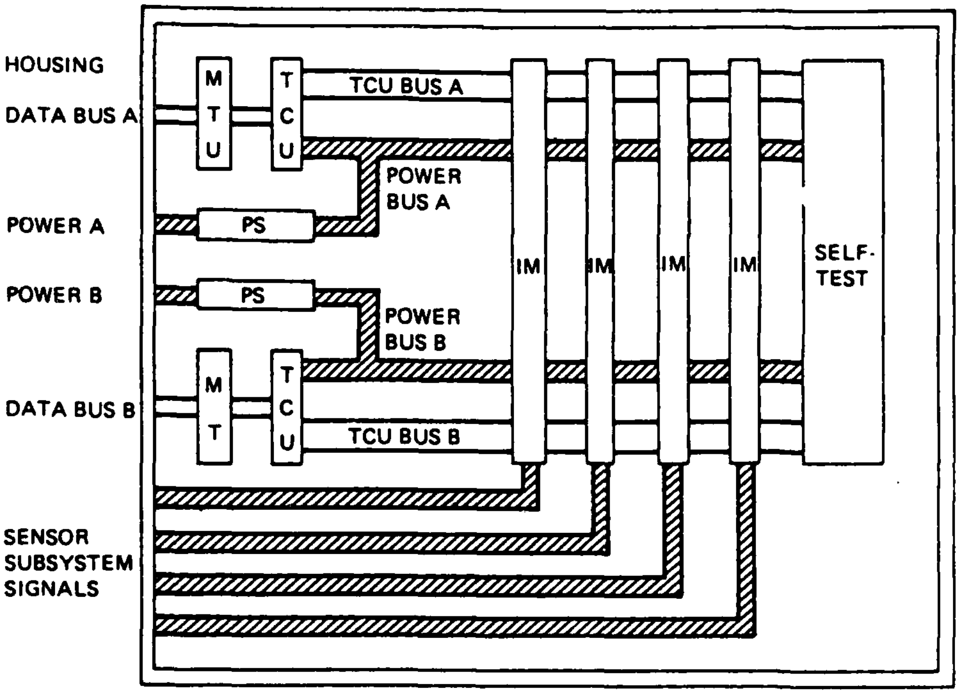


Figure 6.5-4. DAIS Remote Terminal Block Diagram

Interface modules (IM) provide scaling, signal conditioning, and formatting for signals coming from both the data bus and the avionic subsystems. The IM's have a standard TCU interface and thus can be placed in any slot in the RT. This allows the IM mix to be personalized for each RT. As shown in table 6.5-4, interface modules are provided for all classes of electrical interfaces now found in aircraft.

The RT is configured in two 3/4 ATR box sizes: a short RT that is 12.5 in long and a long RT that is 19.5 in. The short RT contains spaces for eight IM's, two MTU pages, and six TCU pages. The long RT contains spaces for 16 IM's, 2 MTU pages, and 6 TCU pages. Both RT's are dual redundant in the MTU and TCU.

Table 6.5-4. DAIS Interface Modules

Input or output modules	No. of channels
Discrete single-ended I/O	32
Discrete differential I/O	16
Switch closure I/O	16
Ac analog I/O	8
Dc analog I/O	8
Serial digital I/O	4
Momentary closure input	16
Synchro input	1
Input and output modules	
Facility I/O	1
Pulse counter/torquer	1

CHAPTER 7
1553 COMMENTARY

Table of Contents

Table of Contents (Continued)

Table of Contents (Continued)

Table of Contents (Continued)

List of Figures

1553B Figures

Table of Contents (Continued)

List of Tables

1553B Tables

7.0 REVIEW AND RATIONALE OF MIL-STD-1553A AND MIL-STD-1553B

This section is an explanation of each part of MIL-STD-1553B on a paragraph-by-paragraph basis. The descriptions include (1) rationale for the requirements specified; (2) the requirements; and (3) identification of differences between MIL-STD-1553A and MIL-STD-1553B. The 1553B part that is discussed is presented first (indented as below), followed by the rationale, explanations, and differences from 1553A.

> 1. SCOPE
>
> 1.1 Scope. This standard establishes requirements for digital, command/response, time division multiplexing (Data Bus) techniques on aircraft. It encompasses the data bus line and its interface electronics illustrated on figure 1, and also defines the concept of operation and information flow on the multiplex data bus and the electrical and functional formats to be employed.
>
> 1.2 Application. When invoked in a specification or statement of work, these requirements shall apply to the multiplex data bus and associated equipment which is developed either alone or as a portion of an aircraft weapon system or subsystem development. The contractor is responsible for invoking all the applicable requirements of this Military Standard on any and all subcontractors he may employ.

Additional sentences were added to 1553B to clarify designer selected options. The basic difference between 1553A and 1553B is that in 1553B the options are defined rather than being left for the user to define as required. It was found that when the standard did not define an item, there was no coordination in its use. Hardware and software had to be redesigned for each new application. Therefore, the one primary goal of 1553B was to provide flexibility without creating new hardware and software designs for each new user. This was accomplished by specifying the "use of" rather than the requirement "to use" in the functional areas and by specifying the electrical interfaces explicitly so that compatibility between designs by different manufacturers could be electrically interchangeable.

> 2. REFERENCED DOCUMENTS
>
> 2.1 Issue of document. The following document, of the issue in effect on date of invitation for bid or request for proposal, forms a part of the standard to the extent specified herein.
>
> SPECIFICATION
>
> MILITARY
>
> MIL-E-6051 Electromagnetic Compatibility Requirements, Systems

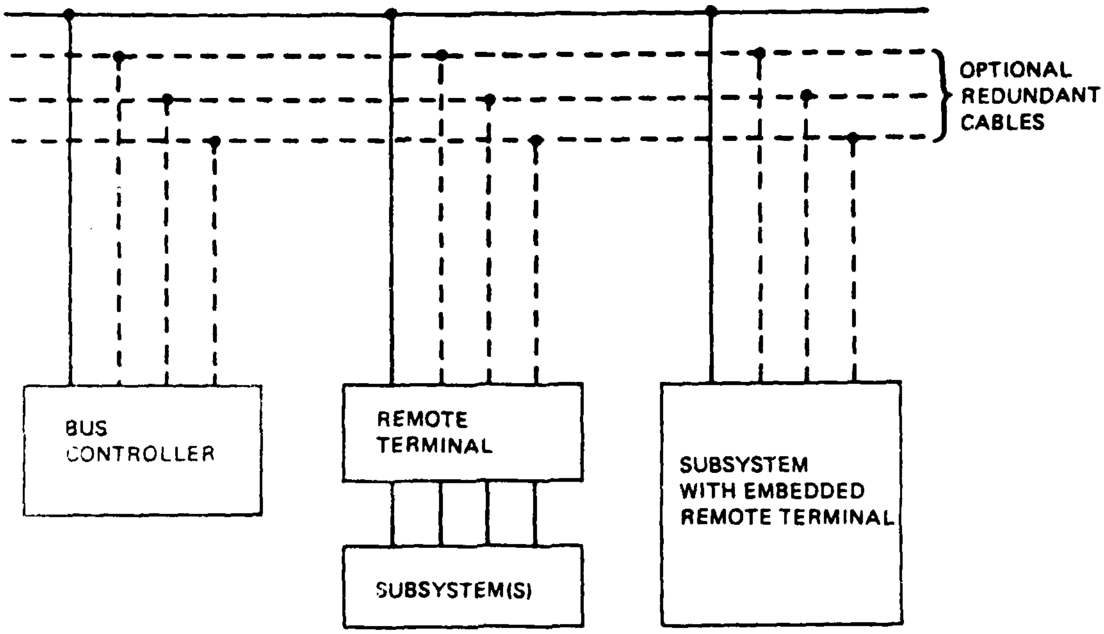


Figure 1 of 1553B. Sample Multiplex Data Bus Architecture

> (Copies of specifications, standards, drawings, and publications required by contractors in connection with specific procurement functions should be obtained from the procuring activity or as directed by the contracting officer.)

The only difference between the two revisions was that in 1553B the references to MIL-STD-461 and MIL-STD-462 were removed (i.e., both electromagnetic interference requirements and measurement standards). MIL-E-6051 "System Electromagnetic Compatibility Requirements" is still applicable in 1553B to define the wiring and cabling provisions of the specification (see MIL-STD-1553B, paragraph 4.5.1.1.5.3).

The definition section of the standard has been expanded and reordered in 1553B. The purpose for the change was to address definitions in order of complexity and to describe new functions, modes, and devices. A comparison of the definition included in 1553A and 1553B is presented in table 7-1.

> 3.1 Bit. Contraction of binary digit: may be either zero or one. In information theory a binary digit is equal to one binary decision or the designation of one or two possible values of states of anything used to store or convey information.
>
> 3.2 Bit rate. The number of bits transmitted per second.

3.3 Pulse code modulation (PCM). The form of modulation in which the modulation signal is sampled, quantized, and coded so that each element of information consists of different types or numbers of pulses and spaces.

3.4 Time division multiplexing (TDM). The transmission of information from several signal sources through one communication system with different signal samples staggered in time to form a composite pulse train.

3.5 Half duplex. Operation of a data transfer system in either direction over a single line, but not in both directions on that line simultaneously.

3.6 Word. In this document a word is a sequence of 16 bits plus sync and parity. There are three types of words: command, status and data.

3.7 Message. A single message is the transmission of a command word, status word, and data words if they are specified. For the case of a remote terminal to remote terminal (RT to RT) transmission, the message shall include the two command words, the two status words, and data words.

3.8 Subsystem. The device or functional unit receiving data transfer service from the data bus.

3.9 Data bus. Whenever a data bus or bus is referred to in this document it shall imply all the hardware including twisted shielded pair cables, isolation resistors, transformers, etc., required to provide a single data path between the bus controller and all the associated remote terminals.

3.10 Terminal. The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate line replaceable units (LRU's) or be contained within the elements of the subsystem.

This definition of terminal is intentionally broad. Terminals in 1553 have common operational characteristics, as well as assigned roles in data bus operation. The three allowable roles are defined in 3.11, 3.12, and 3.13. Common operational requirements of terminals are given in 1553B, paragraph 4.4.1. Note that the definition gives designers complete freedom of functional partitioning of the operating parts of a terminal, and that there is also no restriction of physical partitioning.

3.11 Bus controller. The terminal assigned the task of initiating information transfers on the data bus.

3.12 Bus monitor. The terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time.

Table 7-1. Comparison of MIL-STD-1553A and MIL-STD-1553B Definitions

MIL-STD-1553A definitions (paragraph number)		MIL-STD-1553B definitions (paragraph number)	
Bit	(3.2)	Bit	(3.1)
Bit rate	(3.3)	Bit rate	(3.2)
Pulse code modulation	(3.4)	Pulse code modulation	(3.3)
Time division multiplexing	(3.5)	Time division multiplexing	(3.4)
Half duplex	(3.7)	Half duplex	(3.5)
Word	(3.10)	Word	(3.6)
Message	(3.11)	*Message	(3.7)
–		**Subsystem	(3.8)
Data bus	(3.12)	*Data bus	(3.9)
–		**Terminal	(3.10)
Bus controller	(3.13)	*Bus controller	(3.11)
–		**Bus monitor	(3.12)
Remote terminal	(3.1)	*Remote terminal	(3.13)
Asynchronous operation	(3.8)	Asynchronous operation	(3.14)
Dynamic bus allocation	(3.9)	Dynamic bus control	(3.15)
Command or response mode	(3.6)	Command or response mode	(3.16)
–		**Redundant data bus	(3.17)
–		**Broadcast	(3.18)
–		**Mode code	(3.19)

*Definition changed significantly.
**Not previously defined.

3.13 Remote terminal (RT). All terminals not operating as the bus controller or as a bus monitor.

3.14 Asynchronous operation. For the purpose of this standard, asynchronous operation is the use of an independent clock source in each terminal for message transmission. Decoding is achieved in receiving terminals using clock information derived from the message.

This definition refers to the electrical characteristic by which the timing of message bits in a word are decoded. This use of "asynchronous operation" should not be confused with an asynchronous message that may interrupt or suspend the transmission of synchronous (i.e., periodic) messages in an avionic system.

3.15 Dynamic bus control. The operation of a data bus system in which designated terminals are offered control of the data bus.

3.16 Command/Response. Operation of a data bus system such that remote terminals receive and transmit data only when commanded to do so by the bus controller.

In the case of the definitions for message, bus controller, remote terminal, asynchronous operation, dynamic bus control, and command/response, the change from 1553A to 1553B was developed to produce a more general definition. However, in the definition of data bus, 1553B encompasses more equipment. Instead of including only the wire, the data bus couplers are also included. Two definitions were added for clarity: subsystem and terminal. The others (bus monitor, redundant data bus, broadcast, and mode codes) were added to define the additional requirements stated in 1553B. The function of a bus monitor is to monitor the data bus and record specified bus activity. The objective of defining a bus monitor function is new to 1553B. Two basic capabilities have been identified for the monitor in paragraph 4.4.4 of 1553B: (1) an offline application including flight test recording, maintenance recording, or mission analysis, and (2) a unique data bus terminal, which provides an internal backup bus controller function, with sufficient information to take over as the active bus controller in the event of a switchover or a failure of the active bus controller. In these two roles, the bus monitor hardware may have the performance capability of a terminal (unique address) or may be attached to the data bus without the knowledge of the other bus users (including the bus controller). In this second approach, no bus communication from or to the bus monitor by the bus controller is possible. The bus monitor acts as a passive listener to the specified traffic it is assigned to record. Obviously, the performance of a bus monitor requires the monitoring of the data bus for command words, status words, and data words. From this monitoring, the specific message collection process can occur during normal and abnormal (bus error and recovery) bus traffic. To aid in accomplishing the detection of these words (command and status), the optional instrumentation bits (bit 10 in the status word) and the associated bit in the command word (bit 10) can be set to a logic 1 and a logic 0, respectively.

> 3.17 <u>Redundant data bus</u>. The use of more than one data bus to provide more than one data path between the subsystem, i.e., dual redundant data bus, tri-redundant data bus, etc.

The redundant data bus definition was added to 1553B to identify a particular approach for obtaining multiple data paths to improve message arrival probability. Paragraph 4.6 of 1553B discusses the use of a dual-redundant data bus where the operation is identified as dual standby. In this mode, only one bus is active at any given time, except when superseding commands are sent on the standby bus. Under this condition, the terminal responds to the most recent command.

> 3.18 <u>Broadcast</u>. Operation of a data bus system such that information transmitted by the bus controller or a remote terminal is addressed to more than one of the remote terminals connected to the data bus.

The broadcast definition has been added to 1553B to describe a new protocol option. The use of this protocol allows a bus controller or a remote terminal to address more than one terminal connected to the system. This is accomplished by transmitting a dedicated terminal address (11111) and each receiver withholding the normal status word response.

3.19 Mode code. A means by which the bus controller can communicate with the multiplex bus related hardware, in order to assist in the management of information flow.

The mode code definition was added to 1553B because of the definition of several mode code operations in paragraph 4.3.3.5.1.7. These optional mode codes are used to manage the information transfer system. The basic philosophy of the data bus system is that it is a "transparent data communication link." This means that its operation and management does not involve the use of the sensor data that it is transmitting or receiving. However, overhead is required to manage such a data link. Therefore, command words, status words, and message gaps are required to provide this capability. The combination of command word, mode codes, and responses to these mode codes provide the basis for managing the multiplex system.

4. GENERAL REQUIREMENTS

Several paragraphs have been added, changed, and renumbered in the requirements section of 1553B compared to 1553A.

4.1 Test and operating requirements. All requirements as specified herein shall be valid over the environmental conditions which the multiplex data bus system shall be required to operate.

This new paragraph of the 1553B was added to indicate that the performance requirements specified in this standard shall apply over the environmental conditions in which the multiplex data bus system shall be required to operate. It is anticipated that for most military applications this will be described by MIL-E-5400 Class II and some nuclear-hardening specifications. Because of the diversity of environmental conditions, the standard does not specify these requirements. Therefore, the system designer must determine, from appropriate vehicles or system specifications, the environmental conditions imposed on the multiplex data bus system.

4.2 Data bus operation. The multiplex data bus system in its most elemental configuration shall be as shown on figure 1. The multiplex data bus system shall function asynchronously in a command/response mode, and transmission shall occur in a half-duplex manner. Sole control of information transmission on the bus shall reside with the bus controller, which shall initiate all transmissions. The information flow on the data bus shall be comprised of messages which are, in turn, formed by three types of words (command, data, and status) as defined in 4.3.3.5.

This paragraph is identical to paragraph 4.1 in 1553A with the exclusion of the reference to electromagnetic compatibility, which appears in paragraph 4.5.1.1.5.3 of 1553B.

4.3 Characteristics

4.3.1 Data form. Digital data may be transmitted in any desired form, provided that the chosen form shall be compatible with the message and word formats defined in this

standard. Any unused bit positions in a word shall be transmitted as logic zeroes.

> 4.3.2 <u>Bit priority</u>. The most significant bit shall be transmitted first with the less significant bits following in descending order of value in the data word. The number of bits required to define a quantity shall be consistent with the resolution or accuracy required. In the event that multiple precision quantities (information accuracy or resolution requiring more than 16 bits) are transmitted, the most significant bits shall be transmitted first, followed by the word(s) containing the lesser significant bits in numerical descending order. Bit packing of multiple quantities in a single data word is permitted.

This paragraph is identical to paragraph 4.2 in 1553A with the additional capability in paragraph 4.3.2 of 1553B concerning bit-packing of multiple quantities.

Bit-packing is a method used to improve transmission efficiency when subsystem data, which contain less than 16 bits of information per parameter (word), are collected and distributed in one word or message. Single-bit data and other parameters that are characterized by bit patterns of fewer than 16 bits will not fill the 16 bits of data allowed in 1553 data word format. Two approaches are used to utilize all bits in a word: (1) packing multiple parameters and words and (2) filling in zeros for all unused bits. In the first approach, the encoding and decoding cost must be considered, while in the second approach the inefficiency of sending as little as one bit per word must be considered.

> 4.3.3 <u>Transmission Method</u>

> 4.3.3.1 <u>Modulation</u>. The signal shall be transferred over the data bus in serial digital pulse code modulation form.

This paragraph remained unchanged in both revisions of the standard.

> 4.3.3.2 <u>Data code</u>. The data code shall be Manchester II bi-phase level. A logic one shall be transmitted as a bipolar coded signal 1/0 (i.e., a positive pulse followed by a negative pulse). A logic zero shall be a bipolar coded signal 0/1 (i.e., a negative pulse followed by a positive pulse). A transition through zero occurs at the midpoint of each bit time (see figure 2).

This paragraph remained unchanged in both revisions of the standard.

> 4.3.3.3 <u>Transmission bit rate</u>. The transmission bit rate on the bus shall be 1.0 megabit per second with a combined accuracy and long-term stability of ± 0.1 percent (i.e., ± 1000 Hertz (Hz)). The short-term stability (i.e., stability over 1.0 second interval) shall be at least 0.01 percent (i.e., ± 100 Hz).

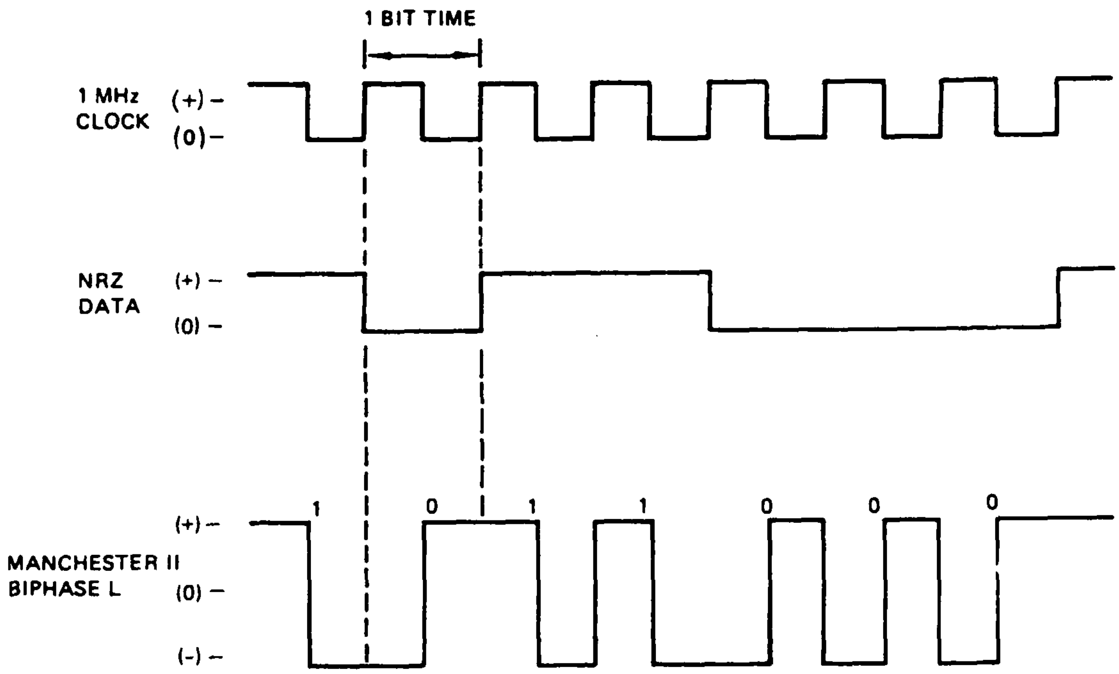


Figure 2 of 1553B. Data Encoding

The long- and short-term stability of the individual internal clocks used to transmit encoded data have been relaxed in 1553B. The order of magnitude reduction in transmission bit rate stability allows for the selection of multiplex bus interface clocks that can meet long-shelf-life requirements of some weapons.

> 4.3.3.4 <u>Word size</u>. The word size shall be 16 bits plus the sync waveform and the parity bit for a total of 20 bits times as shown on figure 3.

The 20-bit word size was selected because it represented the minimum number of bits in a word, when 16 bits of data, a three-bit invalid Manchester sync pattern, and a single parity bit are used. Except for paragraph changes of 4.2.3.4 (1553A) to 4.3.3.4 (1553B) this paragraph remained unchanged. Three-bit invalid Manchester sync pattern is described in 1553B, paragraph 4.3.3.5.1.1. Figure 3 (referenced in this paragraph) is modified in 1553B to reflect the identification of status codes.

> 4.3.3.5 <u>Word formats</u>. The word formats shall be as shown on figure 3 for the command, data, and status words.

Three types of word formats were selected to operate the information transfer system. Each format and the changes reflected in 1553B will be discussed in the following paragraphs.

> 4.3.3.5.1 <u>Command word</u>. A command word shall be comprised of a sync waveform, remote terminal address field,

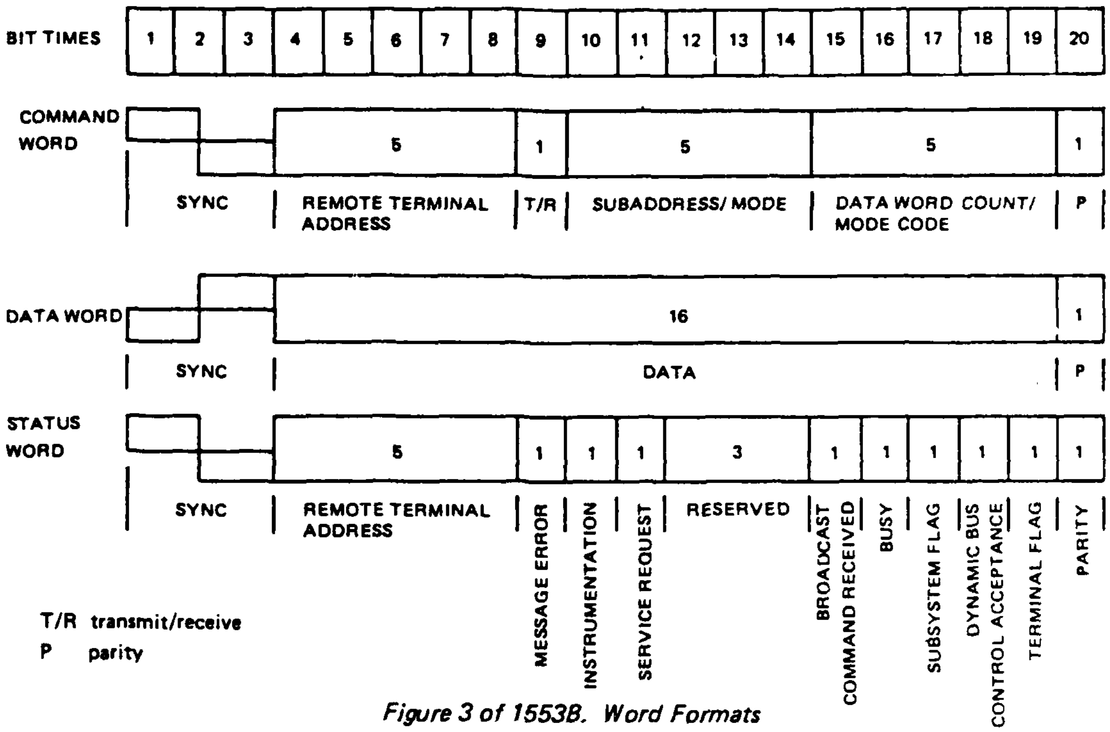


Figure 3 of 1553B. Word Formats

transmit/receive (T/R) bit, subaddress/mode field, word count/mode code field, and a parity (P) bit (see figure 3).

The command word format is used to control and manage the information transfer system. Two basic additions were made to the command word by 1553B. These include the broadcast mode and the identification of the optional mode codes.

4.3.3.5.1.1 Sync. The command sync waveform shall be an invalid Manchester waveform as shown on figure 4. The width shall be three bit times, with the sync waveform being positive for the first one and one-half bit times, and then negative for the following one and one-half bit times. If the next bit following the sync waveform is a logic zero, then the last half of the sync waveform will have an apparent width of two clock periods due to the Manchester encoding.

The sync pattern used in the standard remained unchanged.

4.3.3.5.1.2 Remote terminal address. The next five bits following the sync shall be the RT address. Each RT shall be assigned a unique address. Decimal address 31 (11111) shall not be assigned as a unique address. In addition to its unique address, a RT shall be assigned decimal address 31 (11111) as the common address, if the broadcast option is used.

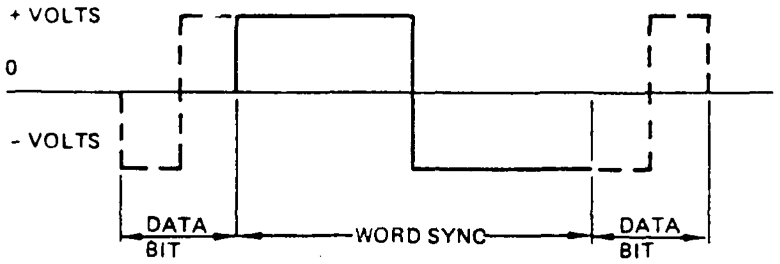


Figure 4 of 1553B. Command and Status Sync

Each remote terminal will be assigned a unique address for which it is responsible to respond when the address is transmitted as part of a command word on the data bus by the active bus controller. Only one remote terminal address cannot be assigned as a unique address (decimal address 31). This address has been assigned to all remote terminals as the common address for which they will receive broadcasted data if the system uses the broadcast option.

The broadcast mode provides a mechanism for transmitting information to multiple users with a single message. The mechanism for accomplishing this was to dedicate address 31 (11111) to be reserved for broadcast messages. Anytime a broadcast message is transmitted, the transmitting terminal will use address 31 rather than a unique terminal address. All other addresses can be assigned as in 1553A. Since multiple users receive a broadcast message, the responding status word must be suppressed. By choosing the address method to accomplish the broadcast mode, all the other formats of the command word are available for use. Broadcast messages can be used with subaddresses and mode codes. The subaddress in a broadcast message can allow multiple users with broadcast reception capability to sort out specific broadcast messages transmitted, if given this capability in hardware or software. Therefore, multiple sets of broadcast messages can be defined. In addition, the broadcast format can be used with mode codes. This allows simultaneous transmission of mode commands to users.

Indiscriminate use of the broadcast technique is not advisable. Designers must question the benefit of discarding the command/response format, in which all message completion failures are known to the bus controller, to the benefits described below. Broadcast use may increase system operation complexity since subaddresses of broadcast address and addressed terminal will not likely be the same. This requires additional subaddresses. Finally, the broadcast technique, if used, adds a failure mode to the system if a terminal in a failure mode used address 31 for a message.

Proper use of the broadcast mode may yield several benefits:

a. Multiple terminals can be communicated with simultaneously, thereby permitting time synchronization of data or commands.

b. Bus duty cycle can be reduced by transmitting data required by multiple users simultaneously instead of sequentially.

c. Some error management can be enhanced by providing a single address by which all terminals can receive commands simultaneously. This permits the bus controller to immediately command a state for the system rather than polling each unit individually with the same command in a serial fashion.

The broadcast message capability can produce considerable reduction in bus usage. This is particularly true for systems using multiple units for redundancy or systems dependent on parallel processing, thus requiring simultaneous data arrival at the processing units. As noted in 1553B, paragraph 10.6 (appendix to 1553B), improper use of the broadcast format can result in undesirable system operation. Since no status word response is allowed from the receiving terminal, discretion must be exercised when applying the capability. To provide message arrival verification, a bit in the status word is set when a valid broadcast message is received. This allows reporting of the reception if requested by the active bus controller using the mode code "transmit status word." In error situations, it may be advisable for the bus controller to request the last command word to verify that the broadcast command was received. There may be situations for which rebroadcast cannot be permitted. Asking for last command first preserves the last status word (i.e., the terminal does not reset or update status). In addition to data transfers, the ability to transmit a broadcast command message provides an effective method for managing the data bus system. This capability is performed using the broadcast address in combination with mode commands.

> 4.3.3.5.1.3 <u>Transmit/receive</u>. The next bit following the remote terminal address shall be the T/R bit, which shall indicate the action required of the RT. A logic zero shall indicate the RT is to receive, and a logic one shall indicate the RT is to transmit.

The transmit/receive bit in the command word indicates the source of data flow in the information transfer system. Basically, the paragraph remained unchanged for both revisions of the standard except for wording changes and paragraph numbering differences.

> 4.3.3.5.1.4 <u>Subaddress/mode</u>. The next five bits following the T/R bit shall be utilized to indicate an RT subaddress or use of mode control, as is dictated by the individual terminal requirements. The subaddress/mode values of 00000 and 11111 are reserved for special purposes, as specified in 4.3.3.5.1.7, and shall not be utilized for any other function.

This field has two functions: (1) the subaddress identification of specific messages to a remote terminal and (2) reserved subaddresses that serve as the identification that a mode command to the information transfer system is being transmitted. Both of these capabilities were present in 1553A. However, an additional mode code designator has been established in 1553B (decimal 31). The use of either 00000 or 11111 in the subaddress/mode field will be decoded to indicate that a mode code command is present in the next five-bit field. This limits the subaddress range to a maximum of 30 unique addresses. If the instrumentation bit (par. 4.3.3.5.3) in the status word is implemented, the subaddresses will be limited to 15 unique subaddresses. The requirements for use of the instrumentation bit are in 1553B, paragraph

4.3.3.5.3.4. In complex remote terminals (i.e., terminals interfacing with several sensors or multiple interface types), the subaddress capacity of a terminal can be exceeded. In addition, messages to a given remote terminal-subaddress may contain "packed" data requiring additional decoding prior to distribution within the terminal. Both of these conditions can cause the remote terminal's design to incorporate a map (i.e., look-up table) approach for subaddress message distribution.

> 4.3.3.5.1.5 <u>Data word count/mode code</u>. The next five bits following the subaddress/mode control shall be the quantity of data words to be either sent out or received by the RT or the optional mode code as specified in 4.3.3.5.1.7. A maximum of 32 data words may be transmitted or received in any one message block. All 1's shall indicate a decimal count of 31, and all 0's shall indicate a decimal count of 32.

The dual function of this field provides for the identification of message lengths for data messages or mode codes for managing the information transfer system. The identification of both of these capabilities was provided in 1553B but only the word count was specified in 1553A (even though the mode code function has remained the same in both revisions). The five-bit field allows 32 data words to be transmitted in a message or 32 specific mode codes. This is accomplished by one data word being represented as 00001, and all zeros being arbitrarily defined as decimal 32.

> 4.3.3.5.1.6 <u>Parity</u>. The last bit in the word shall be used for parity over the preceding 16 bits. Odd parity shall be utilized.

The use of a single parity bit per word was provided to identify bit errors occurring during the transmission and detection of a word. According to the statement in the appendix to 1553B, "Theoretical and empirical evidence indicates that an undetected bit error rate of 10^{-12} can be expected from a practical multiplex system built to this standard." See 1553B, paragraph 10.4. Also see noise test in 1553B, paragraph 4.5.2.1.2.4. This paragraph remained unchanged during both revisions.

> 4.3.3.5.1.7 <u>Optional mode control</u>. For RT's exercising this option a subaddress/mode code of 00000 or 11111 shall imply that the contents of the word count field are to be decoded as a five bit mode command. The mode code shall only be used to communicate with the multiplex bus related hardware, and to assist in the management of information flow, and not to extract data from or feed data to a functional subsystem. Codes 00000 through 01111 shall only be used for mode codes which do not require transfer of a data word. For these codes, the T/R bit shall be set to 1. Codes 10000 through 11111 shall only be used for mode codes which require transfer of a single data word. For these mode codes, the T/R bit shall indicate the direction of data word flow as specified in 4.3.3.5.1.3. No multiple data word transfer shall be implemented with any mode code. The mode codes are reserved for the specific functions as specified in table I and shall not be used for any other purposes. If the designer chooses to implement any of these functions, the

specific codes, T/R bit assignments, and use of a data word, shall be used as indicated. The use of the broadcast command option shall only be applied to particular mode codes as specified in table I.

The basic philosophy of the information transfer system is that it operates as a transparent communication link. "Transparent" means that an application's function does not need to be involved with the management of communication control. Obviously, the information transfer system requires management that introduces overhead into the transmission of data. The command words, status words, status word gaps, and message gaps are the overhead. Within the command word the mode codes provide data bus management capability. The mode codes have been divided into two groups: mode codes without a data word (00000 - 01111) and mode codes with a data word (10000 - 11111). The use of bit 15 in the command word to identify the two types was provided to aid in the decoding process. Also, the use of a single data word instead of multiple data words was adopted to simplify the mode circuitry. Generally, with these two types of mode commands, all management requirements of an information transfer system can be met.

Table I. Assigned Mode Codes

Transmit-receive bit	Mode code	Function	Associated data word	Broadcast command allowed
1	00000	Dynamic bus control	No	No
1	00001	Synchronize	No	Yes
1	00010	Transmit status word	No	No
1	00011	Initiate self-test	No	Yes
1	00100	Transmitter shutdown	No	Yes
1	00101	Override transmitter shutdown	No	Yes
1	00110	Inhibit terminal flag bit	No	Yes
1	00111	Override inhibit terminal flag bit	No	Yes
1	01000	Reset remote terminal	No	Yes
1	01001	Reserved	No	TBD
1	01111	Reserved	No	TBD
1	10000	Transmit vector word	Yes	No
0	10001	Synchronize	Yes	Yes
1	10010	Transmit last command	Yes	No
1	10011	Transmit bit word	Yes	No
0	10100	Selected transmitter shutdown	Yes	Yes
0	10101	Override selected transmitter shutdown	Yes	Yes
1 or 0	10110	Reserved	Yes	TBD
1 or 0	11111	Reserved	Yes	TBD

Note: TBD—to be determined.

Control messages are identified by the subaddress/mode field in the command word being set to 32 (00000) or 31 (11111). (In this case, 1553B defines decimal subaddress 32 to be equal to binary 00000 so that decimal 1 through decimal 31 correspond to binary 00001 through 11111.) All control messages originate with the active bus controller and are received by a single receiver or by multiple receivers (broadcast). A terminal address value of 31 (11111) in the command word indicates a broadcast message, while any other terminal addresses are to identify unique messages to a terminal on the bus. The mode command information is contained completely in the mode code/word field of the command word.

Table I of 1553B should be examined carefully to see the symmetry of the mode codes. The first 16 codes are not transmitted with a data word; the last 16 are. It is not appropriate to broadcast some of the mode codes because of the possibility of bus crashes--simultaneous transmission by two or more terminals. Examples are requests for transmissions from RTs. Also, broadcast of dynamic bus control makes no sense. The T/R bit is important for mode codes 17 to 31 because it defines whether bus controller or RT is to transmit the associated data word.

The use of mode commands option is defined in both versions of the standard; however, 1553B defines each mode command while 1553A only defines dynamic bus control. There is no particular reason for the assignment of the mode codes, except for dynamic bus control (00000), which was previously defined in 1553A, and this separation of mode command by their use of a data word. The purpose of reserved mode commands in each category (with and without data words) is important to allow for controlled expansion of the standard. By controlling the mode code command number and its definition, commonality between various terminals can be maintained. Each mode code command identification is listed in 1553B, table I. All other mode codes are considered illegal commands. The message formats associated with mode commands are shown in figure 7-1.

> 4.3.3.5.1.7.1 <u>Dynamic bus control</u>. The controller shall issue a transmit command to an RT capable of performing the bus control function. This RT shall respond with a status word as specified in 4.3.3.5.3. Control of the data bus passes from the offering bus controller to the accepting RT upon completion of the transmission of the status word by the RT. If the RT rejects control of the data bus, the offering bus controller retains control of the data bus.

The dynamic bus control mode command (00000) is provided to allow the active bus controller a mechanism (using the information transfer system message formats) to offer a potential bus controller (operating as a remote terminal) control of the data bus. Only the single receiver command request (unique address) is allowed to be issued by the active bus controller. The response to this offering of bus controller is provided by the receiving remote terminal using the dynamic bus control acceptance bit in the status word (par. 4.3.3.5.3). Rejection of this request by the remote terminal requires the presently active bus controller to continue offering control to other potential controllers or remain in control. When a remote terminal accepts control of the data bus system by setting the dynamic bus control acceptance bit in the status word, control is relinquished by the presently active bus controller, and the potential bus controller begins bus control.

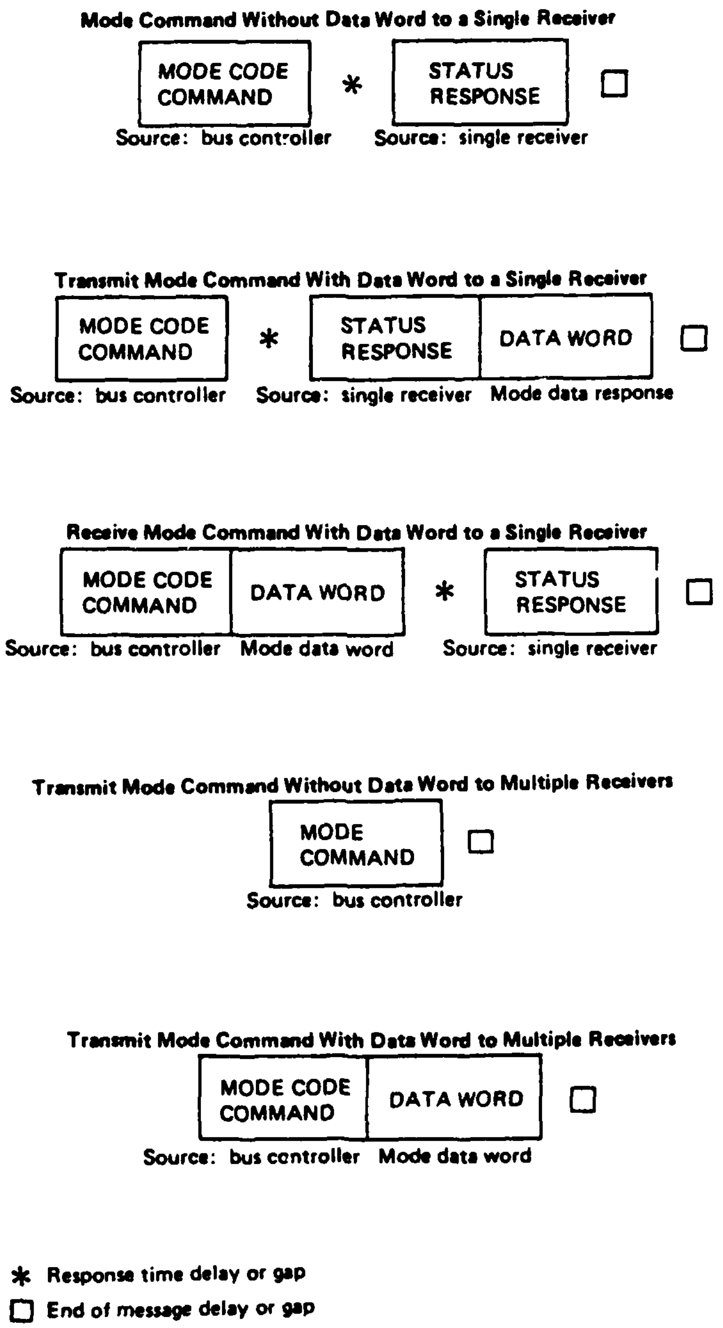


Figure 7-1. Mode Command Message Transfer Formats

Note that the sequence above requires software (or firmware) implementation in all bus controllers.

4.3.3.5.1.7.2 Synchronize (without data word). This command shall cause the RT to synchronize (e.g., to reset the internal timer, to start a sequence, etc.). The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.12 Synchronize (with data word). The RT shall receive a command word followed by a data word as specified in 4.3.3.5.2. The data word shall contain synchronization information for the RT. After receiving the command and data word, the RT shall transmit the status word as specified in 4.3.3.5.3.

Synchronization informs the terminal(s) of an event time to allow coordination between the active bus controller and receiving terminals. Synchronization information may be implicit in the command word (mode code 00001) or a data word (mode code 10001) may be used to follow the command word to provide the synchronization information. If a data word is used, the definition of the bit meanings is the responsibility of the system designer.

4.3.3.5.1.7.3 Transmit status word. This command shall cause the RT to transmit the status word associated with the last valid command word preceding this command. This mode command shall not alter the state of the status word.

The status word associated with mode code (00010) is shown in figure 7-2 and contains the following information:

a. Transmitting terminal address
b. Message error bit
c. Instrumentation bit
d. Service request bit
e. Broadcast command receive bit
f. Busy bit
g. Subsystem flag bit
h. Terminal flag bit

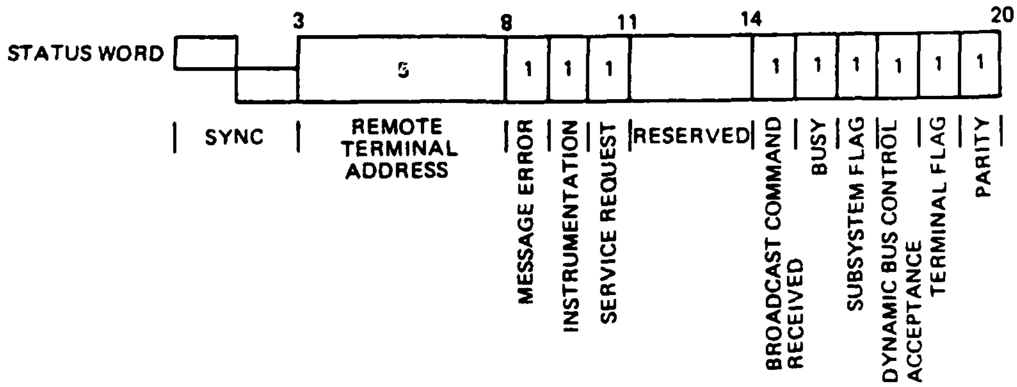


Figure 7-2. Status Word

Details concerning the usage of the status bits are discussed in 1553B, paragraph 4.3.3.5.3. The only message format for acquiring the status word using this mode code is for the bus controller to request the status word from a single receiver. Note that use of this mode code by the bus controller causes the last status word to be transmitted. Some subtle conditions need to be examined by the designer who uses this mode command. For example, if the transmit built-in-test mode command is needed to verify the terminal is operational, that request (see 4.3.3.5.1.7.14) must be issued after the transmit status word mode code to prevent loss of the previous message's status.

> 4.3.3.5.1.7.4 Initiate self test. This command shall be used to initiate self test within the RT. The RT shall transmit the status as specified in 4.3.3.5.3.

The initiate self-test mode command (00011) is provided to initiate built-in-test (BIT) circuitry within remote terminals. The mode code is usually followed, after sufficient time for test completion, by a transmit BIT word mode command yielding the results of the test. The message formats provided for this mode command allow for both individual requests and multiple requests. Notice that the initiate self-test mode command is associated with the multiplex system terminal hardware only.

> 4.3.3.5.1.7.14 Transmit built-in-test (BIT) word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word containing the RT BIT data. This function is intended to supplement the available bits in the status word when the RT hardware is sufficiently complex to warrant its use. The data word, containing the RT BIT data, shall not be altered by the reception of a transmit last command or a transmit status word mode code. This function shall not be used to convey BIT data from the associated subsystem(s).

The transmit BIT word mode command (10011) provides the BIT results available from a terminal, as well as the status word. Typical BIT word information for both embedded and standalone remote terminals includes encoder-decoder failure, analog T/R failures, terminal control circuitry failures, power failures, subsystem interface failures, and protocol errors (e.g., parity, Manchester, word count, status word errors, and status word exceptions). The internal contents of the BIT data word are provided to supplement the appropriate bits already available via the status word for complex terminals. Notice that the transmit BIT word within the remote terminal "...shall not be altered by the reception of a transmit last command or transmit status word mode code" received by the terminal. This allows error handling and recovery procedures to be used without changing the error data recorded in this word. However, the RT will only save the last command, and the status code field (of the status word) will not be changed if transmit last command or transmit status word mode commands are transmitted. If, however, any other transmissions are made to the RT, the status code field may change (e.g., if a message error occurred during the transmission). Broadcast of this command by the bus controller is not allowed. See paragraphs 4.3.3.5.1.7.3 and 4.3.3.5.1.7.13.

Another point worth noting is that the function of transmitting RT BIT data "... shall not be used to convey BIT data from the associated subsystem(s)." Subsystem fault investigation, when indicated by the subsystem flag, is not specified or otherwise restricted by 1553. Therefore, system designers must make the necessary provisions.

> 4.3.3.5.1.7.5 <u>Transmitter shutdown</u>. This command (to only be used with dual redundant bus systems) shall cause the RT to disable the transmitter associated with the redundant bus. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.
>
> 4.3.3.5.1.7.6 <u>Override transmitter shutdown</u>. This command (to only be used with dual redundant bus system) shall cause the RT to enable a transmitter which was previously disabled. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.
>
> 4.3.3.5.1.7.15 <u>Selected transmitter shutdown</u>. This command shall cause the RT to disable the transmitter associated with a specified redundant data bus. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be disabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.
>
> 4.3.3.5.1.7.16 <u>Override selected transmitter shutdown</u>. This command shall cause the RT to enable a transmitter which was previously disabled. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be enabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.

Four mode code commands are provided to control transmitters associated with terminals in a system. These commands can be sent to a single receiver or broadcast to multiple users.

The transmitter shutdown mode code (00100) is used in a dual-redundant bus structure where the command causes the transmitter associated with the other redundant bus to terminate transmissions. No data word is provided for this mode.

The override transmitter shutdown mode code (00101) is used in a dual-redundant bus structure where the command allows the transmitter

previously disabled associated with the redundant bus to transmit when commanded by a normal bus command initiated by the active bus controller. No data word is provided for this mode code.

The selected transmitter shutdown mode code (10100) is used in a multiple (greater than two) redundant bus structure where the command causes the selected transmitter to terminate transmissions on its bus. A data word is used to identify the selected transmitter.

The override selected transmitter shutdown mode code (10101) is used in a multiple (greater than two) redundant bus structure where the command allows the selected transmitter to transmit on its bus when commanded by a normal bus command initiated by the active bus controller. A data word is used to identify the selected transmitter.

4.3.3.5.1.7.7 Inhibit terminal flag (T/F) bit. This command shall cause the RT to set the T/F bit in the status word specified in 4.3.3.5.3 to logic zero until otherwise commanded. The RT shall transmit the status word as specified in 4.3.3.5.3.

The inhibit terminal flag mode code (00110) is used to set the terminal flag bit in the status word to an unfailed condition regardless of the actual state of the terminal being addressed. This mode code is primarily used to prevent continued interrupts to the error handling and recovery system when the failure has been noted and the system reconfigured as required. Commanding this mode code prevents future failures from being reported, which normally would be reported using the terminal flag in each subsequent status word response. The message format associated with the mode code allows for both single receivers and multiple receivers to respond. No data word is required with this mode code. Note that the terminal flag, which is used to indicate an RT fault condition is implicitly limited to terminal faults.

4.3.3.5.1.7.8 Override inhibit T/F bit. This command shall cause the RT to override the inhibit T/F bit specified in 4.3.3.5.1.7.7. The RT shall transmit the status word as specified in 4.3.3.5.3.

The override inhibit T/F flag mode command (00111) negates the inhibit function thus allowing the T/F flag bit in the status response to report present condition of the terminal. This mode code can be transmitted by the active bus controller to both single and multiple receivers. There is no data word associated with this mode code.

4.3.3.5.1.7.9 Reset remote terminal. This command shall be used to reset the RT to a power up initialized state. The RT shall first transmit its status word, and then reset.

The reset remote terminal mode code (01000) causes the addressed terminal to reset itself to a power-up initialized state. This mode code may be transmitted to an individual or to multiple terminals.

4.3.3.5.1.7.11 Transmit vector word. This command shall cause the RT to transmit a status word as specified in 4.3.3.5.3 and a data word containing service request information.

The transmit vector word mode code (10000) is associated with the service request bit in the status word and is used to determine specific service being required by the terminal. The service request bit and the transmit vector word provide the only means available for the terminal to request the scheduling of an asynchronous message if more than one service request exists per terminal. The message format for this single receiver operation contains a data word associated with the terminal's response. Figure 7-3 illustrates the message formats associated with this mode command.

4.3.3.5.1.7.13 Transmit last command word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word which contains bits 4-19 of the last command word, excluding a transmit last command word mode code received by the RT. This mode command shall not alter the state of the RT's status word.

The transmit last command mode code (10010) is used in the error handling and recovery process to determine the last valid command received by the terminal, except for this mode code. Also this mode code will not change the state of the status word. The message format associated with the single receiver last command word contains a data word from the responding terminal. The data word contains the previous 16 bits of the last valid command word received. Notice that this mode command will not alter the state of the receiving terminals status word. This fact allows this mode command to be used in error handling and recovery operation without affecting the status word, which can have added error data.

4.3.3.5.1.7.10 Reserved mode codes (01001 to 01111). These mode codes are reserved for future use and shall not be used.

4.3.3.5.1.7.17 Reserved mode codes (10110 to 11111). These mode codes are reserved for future use and shall not be used.

Each of the mode code types (with and without data words) have several unused mode codes that are reserved for future use and cannot be used without the permission of the Military Standard's Controlling Agency.

4.3.3.5.2 Data word. A data word shall be comprised of a sync waveform, data bits, and a parity bit (see figure 3).

Figure 7-4 illustrates the 1553 data word.

4.3.3.5.2.1 Sync. The data sync waveform shall be an invalid Manchester waveform as shown on figure 5. The width shall be three bit times, with the waveform being negative for the first one and one-half bit times, and then positive for the following one and one-half bit times. Note that if the bits preceding and following the sync are logic ones, then the apparent width of the sync waveform will be increased to four bit times.

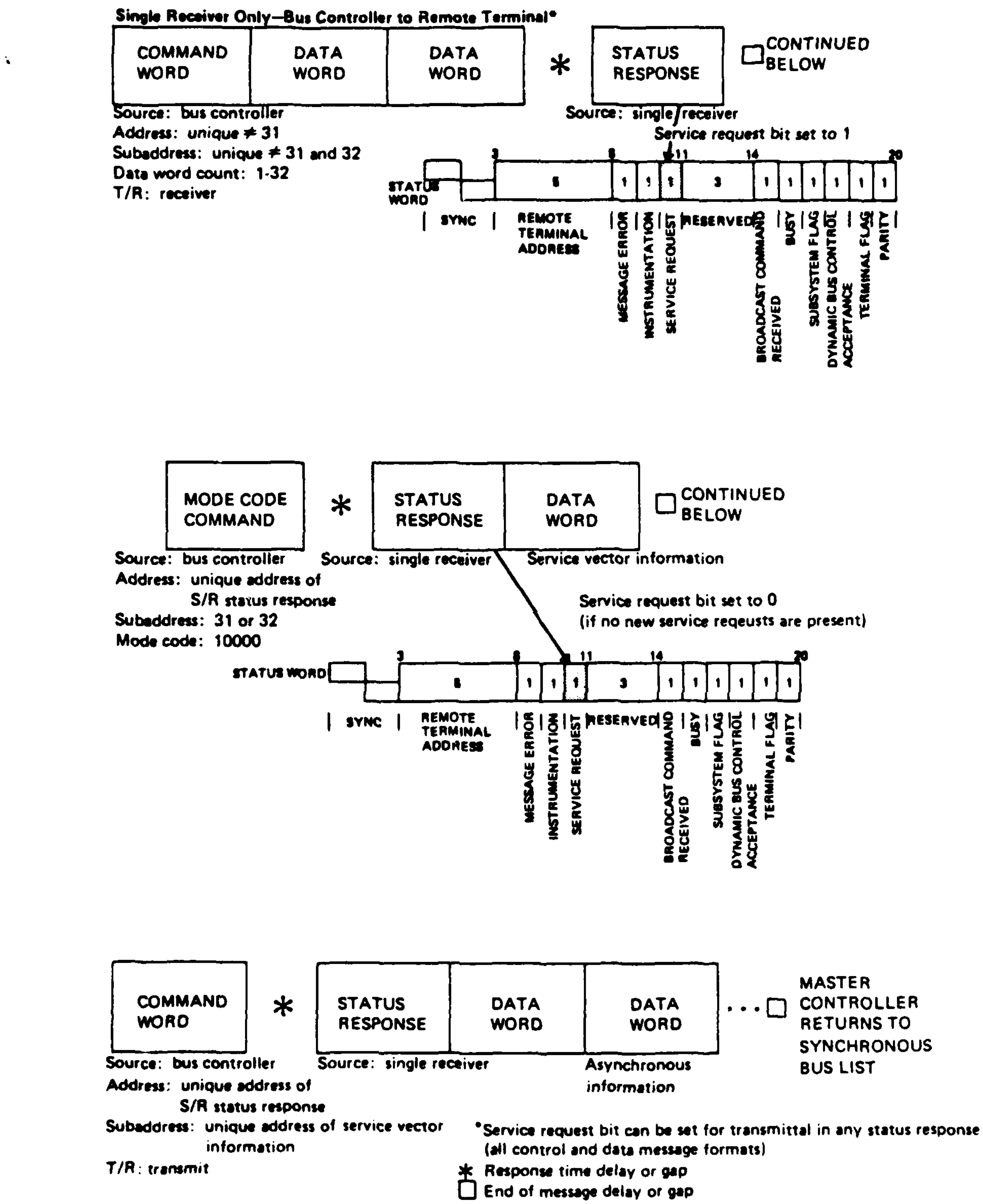


Figure 7-3. Transmit Vector Word Transfer Format

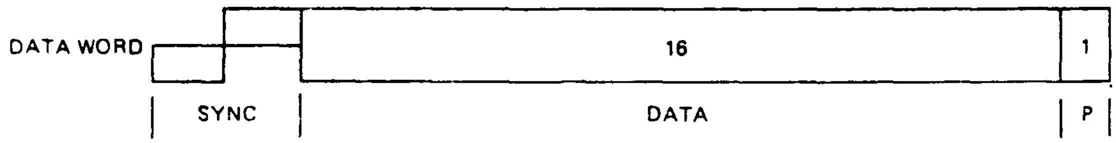


Figure 7-4. 1553 Data Word

4.3.3.5.2.2 Data. The sixteen bits following the sync shall be utilized for data transmission as specified in 4.3.2.

4.3.3.5.2.3 Parity. The last bit shall be utilized for parity as specified in 4.3.3.5.1.6.

Data words are used to transmit parameter data, which is the goal of the information transfer system. Data words are distinguished from command and status words by the inverted three-bit sync pattern. Both packed and unpacked data may be transmitted in the 16-bit data field. Odd parity on the data field provides data integrity identical to the command and status word formats. No changes in the 1553A or 1553B have occurred in these paragraphs except for paragraph numbering (e.g., 4.2.3.5.2 for 1553A and 4.3.3.5.2 for 1553B).

4.3.3.5.3 Status word. A status word shall be comprised of a sync waveform, RT address, message error bit, instrumentation bit, service request bit, three reserved bits, broadcast command received bit, busy bit, subsystem flag bit, dynamic bus control bit, terminal flag bit, and a parity bit. For optional broadcast operation, transmission of the status word shall be suppressed as specified in 4.3.3.6.7.

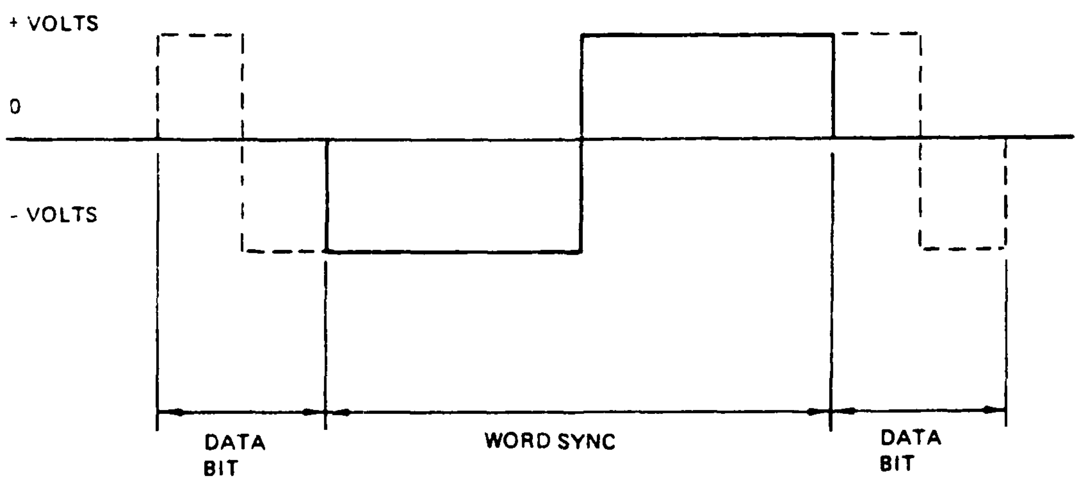


Figure 5 of 1553B. Data Sync

4.3.3.5.3.1 Sync. The status sync waveform shall be as specified in 4.3.3.5.1.1.

4.3.3.5.3.2 RT address. The next five bits following the sync shall contain the address of the RT which is transmitting the status word as defined in 4.3.3.5.1.2.

The status word is part of the basic overhead requirements of the data bus system. The status word is shown in figure 7-5 and is divided into the following fields:

a. Sync (same as command sync)
b. Terminal address
c. Status field
d. Parity (P)

The five-bit address field identifies the transmitting terminal's address, while the remote terminal's status is based on bits set in the status field. The status field consists of the following information:

a. Message error bit
b. Instrumentation bit
c. Service request bit
d. Reserved field
e. Broadcast command received bit
f. Busy bit
g. Subsystem flag
h. Dynamic bus control acceptance bit
i. Terminal flag

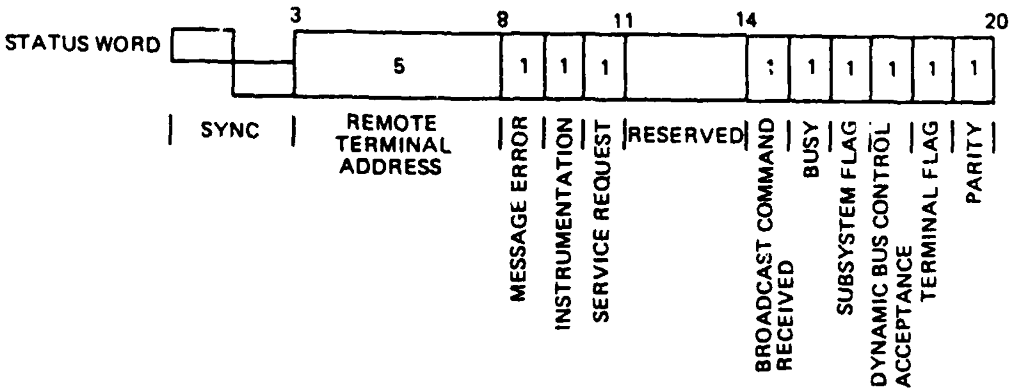


Figure 7-5. Status Word

4.3.3.5.3.3 Message error bit. The status word bit at bit time nine (see figure 3) shall be utilized to indicate that one or more of the data words associated with the preceding receive command word from the bus controller has failed to pass the RT's validity tests as specified in 4.4.1.1. This bit shall also be set under the conditions specified in 4.4.1.2, 4.4.3.4 and 4.4.3.6. A logic one shall indicate the presence of a message error, and a logic zero shall show its absence. All RT's shall implement the message error bit.

The message error bit is set to logic one to indicate that one or more of the data words associated with the preceding received message has failed to pass the message validity test. The message validity requirements are:

a. Word validation--word begins with valid sync, Manchester II code correctly transmitted, 16 data bits plus parity, and word parity odd

b. Contiguous words within a message

c. Address validation--matches address unique terminal or broadcast address

d. Illegal command--a terminal with the illegal command detection circuitry detects an illegal command

The status word will be transmitted if the message validity requirements are met (see para. 4.4.3.5 and 4.4.3.6). When a message error occurs in a broadcast message format, the message error bit will be set in the status word and the status response withheld as required by broadcast message format.

4.3.3.5.3.4 Instrumentation bit. The status word bit time of 10 (see figure 3) shall be reserved for the instrumentation bit and shall always be a logic zero. This bit is intended to be used in conjunction with a logic one in bit time 10 of the command word to distinguish between a command word and a status word. The use of the instrumentation bit is optional.

The instrumentation bit in the status field is set to distinguish the status word from the command word. Since the sync field (three bits) is used to distinguish the command and status words from a data word, a mechanism to distinguish command and status is provided by the instrumentation bit. By setting this bit to logic zero for all conditions and setting the same bit position in the command word to a logic one, the command and status words are identifiable. If used, this approach reduces the possible subaddress in the command word to 15 and requires subaddress 31 (11111) to be used to identify mode commands (both 31 and 32 are allowed). If not used, the bit will remain set to logic zero in the status word for all conditions.

4.3.3.5.3.5 Service request bit. The status word bit at bit time eleven (see figure 3) shall be reserved for the service request bit. The use of this bit is optional. This bit when used, shall indicate the need for the bus controller to take specific predefined actions relative to either the RT or associated subsystem. Multiple subsystems, interfaced to a

single RT, which individually require a service request signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific requesting subsystem. The service request bit is intended to be used only to trigger data transfer operations which take place on an exception rather than periodic basis. A logic one shall indicate the presence of a service request, and a logic zero its absence. If this function is not implemented, the bit shall be set to zero.

The service request bit is provided to indicate to the active bus controller that a remote terminal requests service. When this bit in the status word is set to logic one, the active bus controller may take a predetermined action or use mode command (transmit vector word) to identify the specific request. The message format for acquiring this is discussed under transmit vector word mode command (see fig. 7-3).

4.3.3.5.3.6 Reserved status bits. The status word bits at bit times 12 through 14 are reserved for future use and shall not be used. These bits shall be set to a logic zero.

The three bit-field (12-14) is reserved for future requirements and is set to logic zero. Any bit in this field not set to logic zero will be disregarded.

4.3.3.5.3.7 Broadcast command received bit. The status word at bit time 15 shall be set to a logic one to indicate that the preceding valid command word was a broadcast command and a logic zero shall show it was not a broadcast command. If the broadcast command option is not used, this bit shall be set to a logic zero.

The broadcast command received bit is set to logic one when the preceding valid command word was a broadcast command (address 31). Since broadcast message formats require the receiving remote terminals to suppress their status words, the broadcast command received bit is set to identify that the command was received properly. If the broadcast message validity is desired, the message format shown in figure 7-6 is used to determine this information. The broadcast command received bit will be reset when the next valid command is received by the remote terminal, unless the next valid command is transmit status word or transmit last command.

4.3.3.5.3.8 Busy bit. The status word bit at bit time 16 (see figure 3) shall be reserved for the busy bit. The use of this bit is optional. This bit, when used, shall indicate that the RT or subsystem is unable to move data to or from the subsystem in compliance with the bus controller's command. A logic one shall indicate the presence of a busy condition, and a logic zero its absence. In the event the busy bit is set in response to a transmit command, then the RT shall transmit its status word only. If this function is not implemented, the bit shall be set to logic zero.

Multiple Receivers—Bus Controller to Remote Terminals

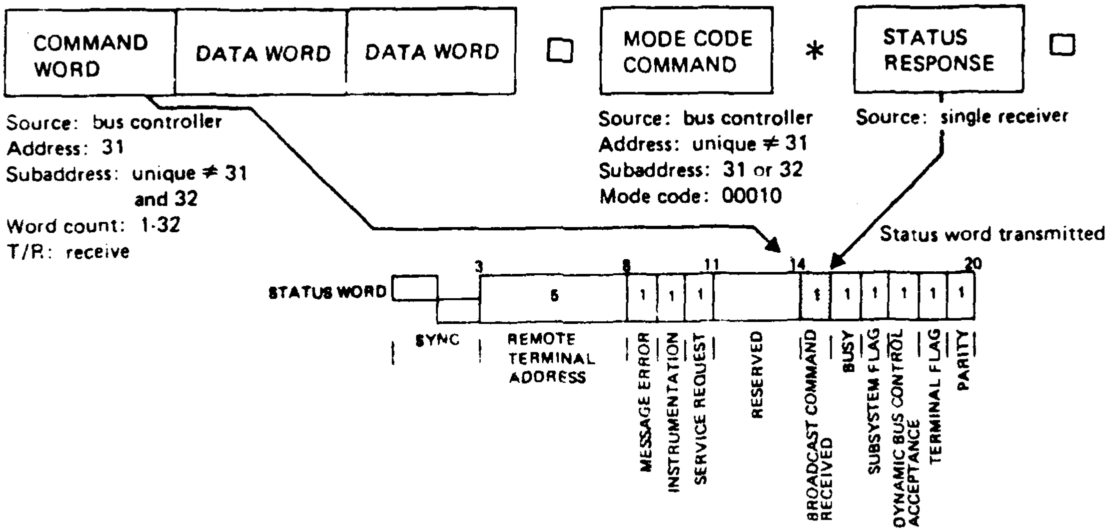


Figure 7-6. Broadcast Command Receive Bit

The busy bit in the status word is set to logic one to indicate to the active bus controller that the remote terminal is unable to move data to or from the subsystem in compliance with the bus controller's command. The message format associated with a busy condition is shown in figure 7-7. A busy condition can exist within a remote terminal at any time causing it to be nonresponsive to a command to send data or to be unable to receive data. This condition can exist for all message formats. In each case except the broadcast message formats, the active bus controller will determine the busy condition immediately upon status response. In the case of the broadcast message formats, this information will not be known unless the receiving terminals are polled after the broadcast message requesting their status. If the status word has the broadcast received bit set, the message was received and the terminal was not busy.

> 4.3.3.5.3.9 <u>Subsystem flag bit</u>. The status word bit at bit time 17 (see figure 3) shall be reserved for the subsystem flag bit. The use of this bit is optional. This bit, when used, shall flag a subsystem fault condition, and alert the bus controller to potentially invalid data. Multiple subsystems, interfaced to a single RT, which individually require a subsystem flag bit signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific reporting subsystem. A logic one shall indicate the presence of the flag, and a logic zero its absence. If not used, this bit shall be set to logic zero.

Single Receiver—Bus Controller to Remote Terminal

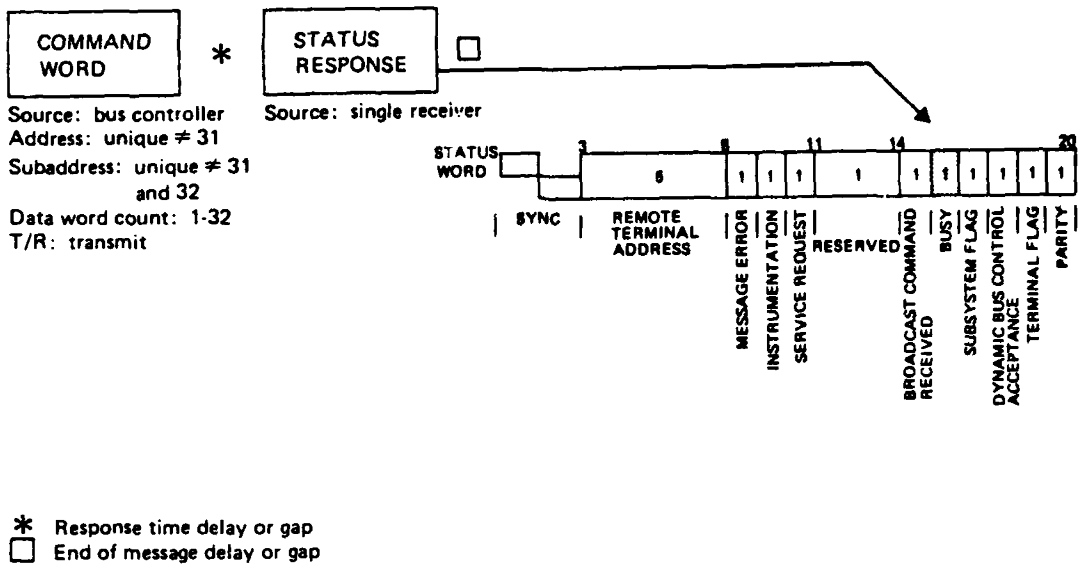


Figure 7-7. Busy Bit

The subsystem flag bit is provided to indicate to the active bus controller that a subsystem fault condition exists and that data being requested from the subsystem may be invalid. The subsystem flag may be set in any transmitted status word.

> 4.3.3.5.3.10 Dynamic bus control acceptance bit. The status word bit at bit time 18 (see figure 3) shall be reserved for the acceptance of dynamic bus control. This bit shall be used if the RT implements the optional dynamic bus control function. This bit, when used, shall indicate acceptance or rejection of a dynamic bus control offer as specified in 4.3.3.5.1.7.1. A logic one shall indicate acceptance of control, and a logic zero shall indicate rejection of control. If this function is not used, this bit shall be set to logic zero.

This bit is provided to indicate the acceptance of the bus controller offer by the active bus controller to become the next bus controller. The offer of bus control occurs when the presently active bus controller has completed its established message list and issues a dynamic bus control mode command to the remote terminal that is to be the next potential controller. To accept the offer the potential bus controller sets its dynamic bus control acceptance bit in the status word and transmits the status word. The establishment of who the next potential controller will be is a system issue.

> 4.3.3.5.3.11 Terminal flag bit. The status word bit at bit time 19 (see figure 3) shall be reserved for the terminal flag function. The use of this bit is optional. This bit,

> when used, shall flag a RT fault condition. A logic one shall indicate the presence of the flag, and a logic zero, its absence. If not used, this bit shall be set to logic zero.

The terminal flag bit is set to a logic one to indicate a fault within the remote terminal. This bit is used in connection with three mode code commands.

a. Inhibit T/F flag
b. Override inhibit T/F flag
c. Transmit BIT word

The first two mode code commands deactivate and activate the functional operation of the bit. The transmit BIT word mode code command is used to acquire more detailed information about the terminal's failure.

> 4.3.3.5.3.12 Parity bit. The least significant bit in the status word shall be utilized for parity as specified in 4.3.3.5.1.6.

The use of a single parity bit per word was provided to identify any bit errors occurring during the transmission and detection of a word. This odd parity check will detect an odd number of bit errors occurring in a word. This requirement produces an undetected bit error rate of 10^{-12}, which was considered satisfactory for a general-purpose information transfer system. This *paragraph remained unchanged during both revisions. See also* 1553B, paragraph 4.3.3.5.1.6.

> 4.3.3.5.4 Status word reset. The status word bit, with the exception of the address, shall be set to logic zero after a valid command word is received by the RT with the exception as specified in 4.3.3.5.1.7. If the conditions which caused bits in the status word to be set (e.g., terminal flag) continue after the bits are reset to logic zero, then the affected status word bit shall be again set, and then transmitted on the bus as required.

This paragraph was added to 1553B to clarify the hardware requirements associated with resetting the status code field of the status word. Figure 7-5 shows the status word and the information available in this field.

One reason for the reset definition is to provide--

a. The ability to obtain the latest status information of the remote terminal: this prevents conditions from being reported for longer than they actually exist.

b. The ability to obtain the status code analysis of the previous results of a valid command: this allows an orderly error handling and recovery approach to be accomplished by the bus controller with the information associated with error analysis data contained within this field or other data associated within the RT (e.g., last command word and BIT).

The second reason for obtaining the status code field not reset was to allow error recovery using two mode codes of paragraph 4.3.3.5.1.7. Even though all mode codes are referenced in the status word reset paragraph, only two are required to retain the last status word in the terminal:

a. Transmit status word
b. Transmit last command word

In other words, all other valid messages received, including mode commands, will allow the RT to reset the status word, except these two.

Both of these mode codes can be transmitted to the RT without changing the bits in the status code field of the last valid command word in question. Therefore, it is essential that an error recovery procedure be established for the bus controller that takes into account (1) the ability of the RT hardware to collect error data, (2) the format of the data that must be requested by the bus controller to prevent data lost, and (3) the ability of the bus controller hardware and software to receive and react to these data. As many as three mode codes may be involved in this process:

a. Transmit last command
b. Transmit status word
c. Transmit BIT word

> 4.3.3.6 Message formats. The messages transmitted on the data bus shall be in accordance with the formats on figure 6 and figure 7. The maximum and minimum response times shall be as stated in 4.3.3.7 and 4.3.3.8. No message formats, other than those defined herein, shall be used on the bus.

The 1553B section of the standard contains two additional message format descriptions that are not contained in revision A. One of these is an explanation of the optional mode code message format that is allowed in revision A but not described with message format diagrams. The other description is the message formats associated with the optional broadcast protocol. The command/response protocol provides two types of message formats:

a. Data messages
b. Control messages

> 4.3.3.6.1 Bus controller to remote terminal transfers. The bus controller shall issue a receive command followed by the specified number of data words. The RT shall, after message validation, transmit a status word back to the controller. The command and data words shall be transmitted in a contiguous fashion with no interword gaps.

> 4.3.3.6.2 Remote terminal to bus controller transfers. The bus controller shall issue a transmit command to the RT. The RT shall, after command word validation, transmit a status word back to the bus controller, followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no interword gaps.

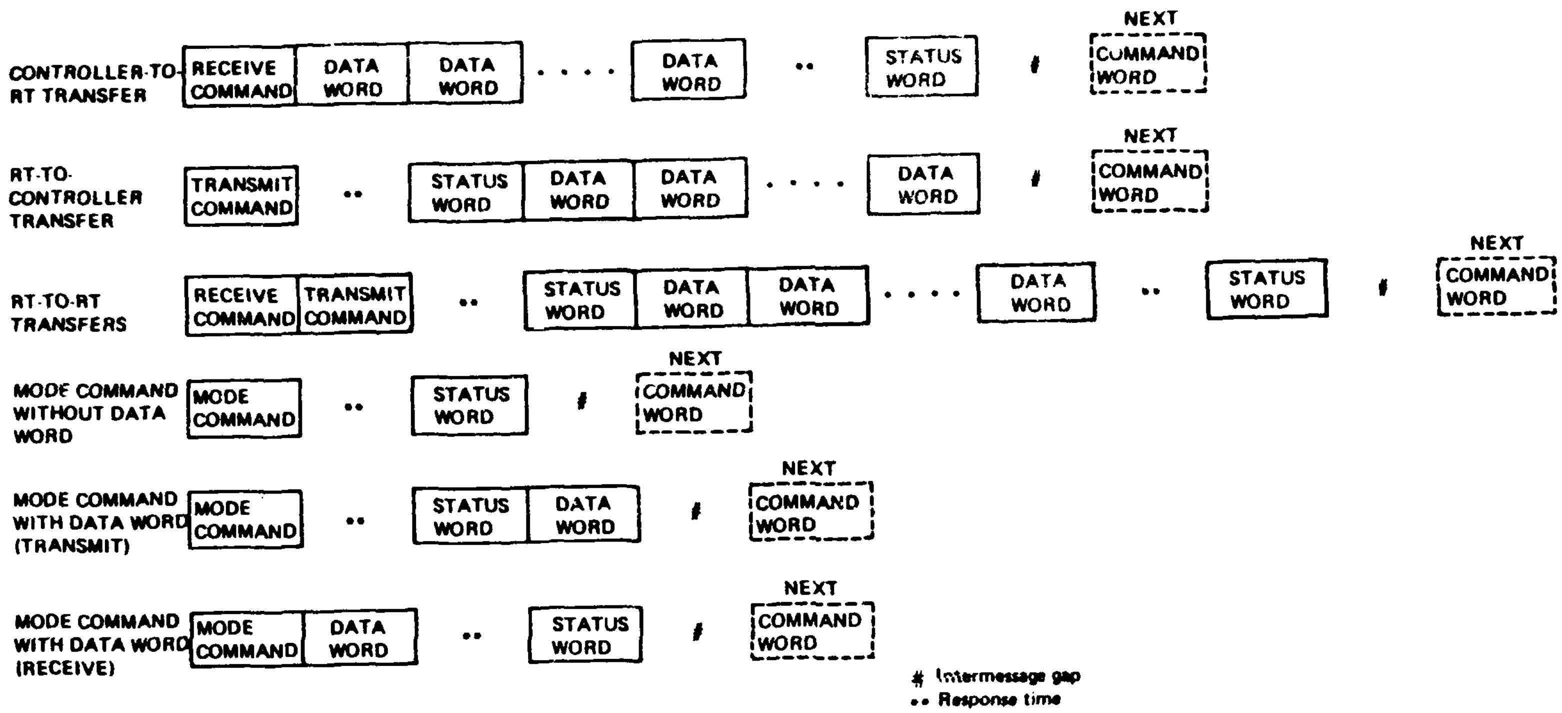


Figure 6 of 1553B. Information Transfer Formats

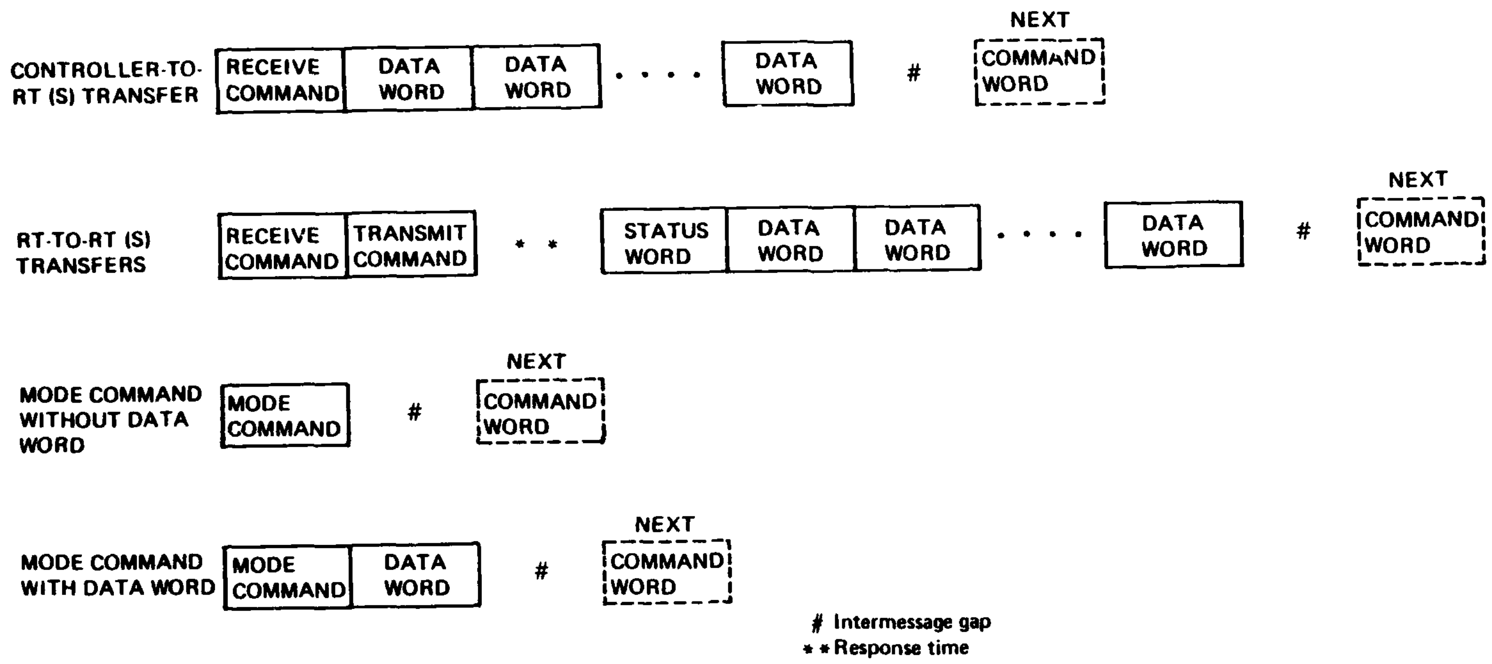


Figure 7 of 1553B. Broadcast Information Transfer Formats

4.3.3.6.3 <u>Remote terminal to remote terminal transfers</u>. The bus controller shall issue a receive command to RT A followed contiguously by a transmit command to RT B. RT B shall, after command verification, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. At the conclusion of the data transmission by RT B, RT A shall transmit a status word within the specified time period.

4.3.3.6.7 <u>Optional broadcast command</u>. See 10.6 for additional information on the use of the broadcast command.

4.3.3.6.7.1 <u>Bus controller to remote terminal(s) transfer (broadcast)</u>. The bus controller shall issue a receive command word with 11111 in the RT address field followed by the specified number of data words. The command word and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message validation, set the broadcast command received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.2 <u>Remote terminal to remote terminal(s) transfers (broadcast)</u>. The bus controller shall issue a receive command word with 11111 in the RT address field followed by a transmit command to RT A using the RT's address. RT A shall, after command word validation, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option, excluding RT A, shall after message validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

Data messages are used to communicate subsystem data to meet the purpose of the integration. As in the control messages, there are two message types: single receiver and multiple receiver messages. These are transmitted in the following manner:

Single receiver

a. Bus controller to remote terminal
b. Remote terminal to bus controller
c. Remote terminal to remote terminal

Multiple receivers

a. Bus controller to multiple remote terminals
b. Remote terminal to multiple remote terminals

Each of these messages is transmitted using command and status words for control operation. The command word is used to--

a. Identify the receiving terminal(s)

b. Identify if data are to be received or transmitted by the receiving terminal(s)

c. Identify the specific message identification (subaddress) within the remote terminal(s)

d. Notify the terminal(s) of the number of data words to be received or transmitted

The command word, status word and data word format for accomplishing these messages are described in paragraphs 4.3.3.5.1, 4.3.3.5.2, and 4.3.3.5.3. Using these words, the format for data message transmissions is developed. The single receiver data message formats are shown in figure 7-8. The message formats associated with multiple receiving terminals are shown in figure 7-9.

4.3.3.6.4 Mode command without data word. The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word.

4.3.3.6.5 Mode command with data word (transmit). The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word followed by one data word. The status word and data word shall be transmitted in a contiguous fashion with no gap.

4.3.3.6.6 Mode command with data word (receive). The bus controller shall issue a receive command to the RT using a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT shall, after command and data word validation, transmit a status word back to the controller.

4.3.3.6.7 Optional broadcast command. See 10.6 for additional information on the use of the broadcast command.

4.3.3.6.7.3 Mode command without data word (broadcast). The bus controller shall issue a transmit command word with 11111 in the RT address field, and a mode code specified in table I. The RT(s) with the broadcast option shall after command word validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.4 Mode command with data word (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field and a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message

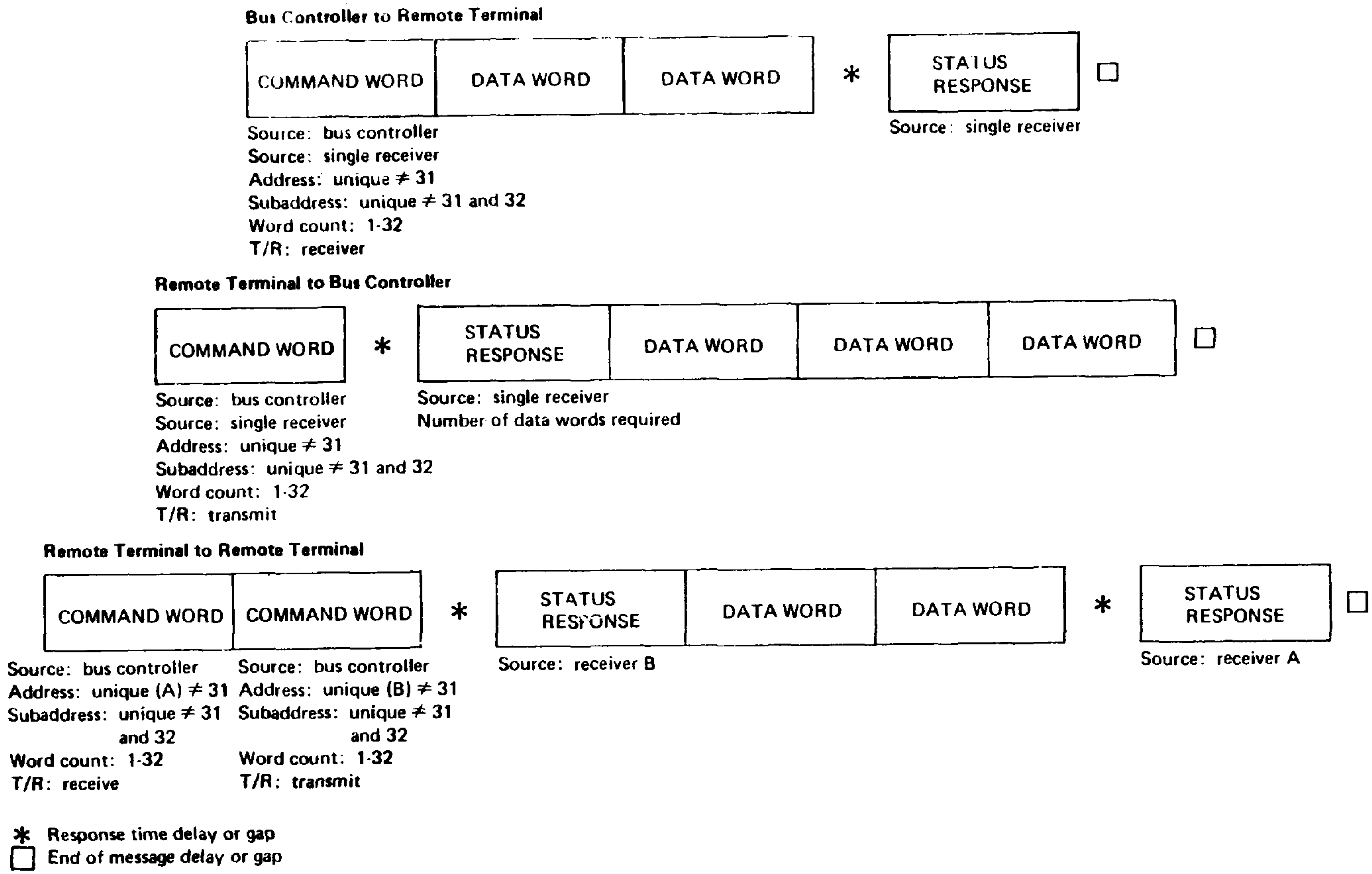


Figure 7-8. Single-Receiver Data Message Formats

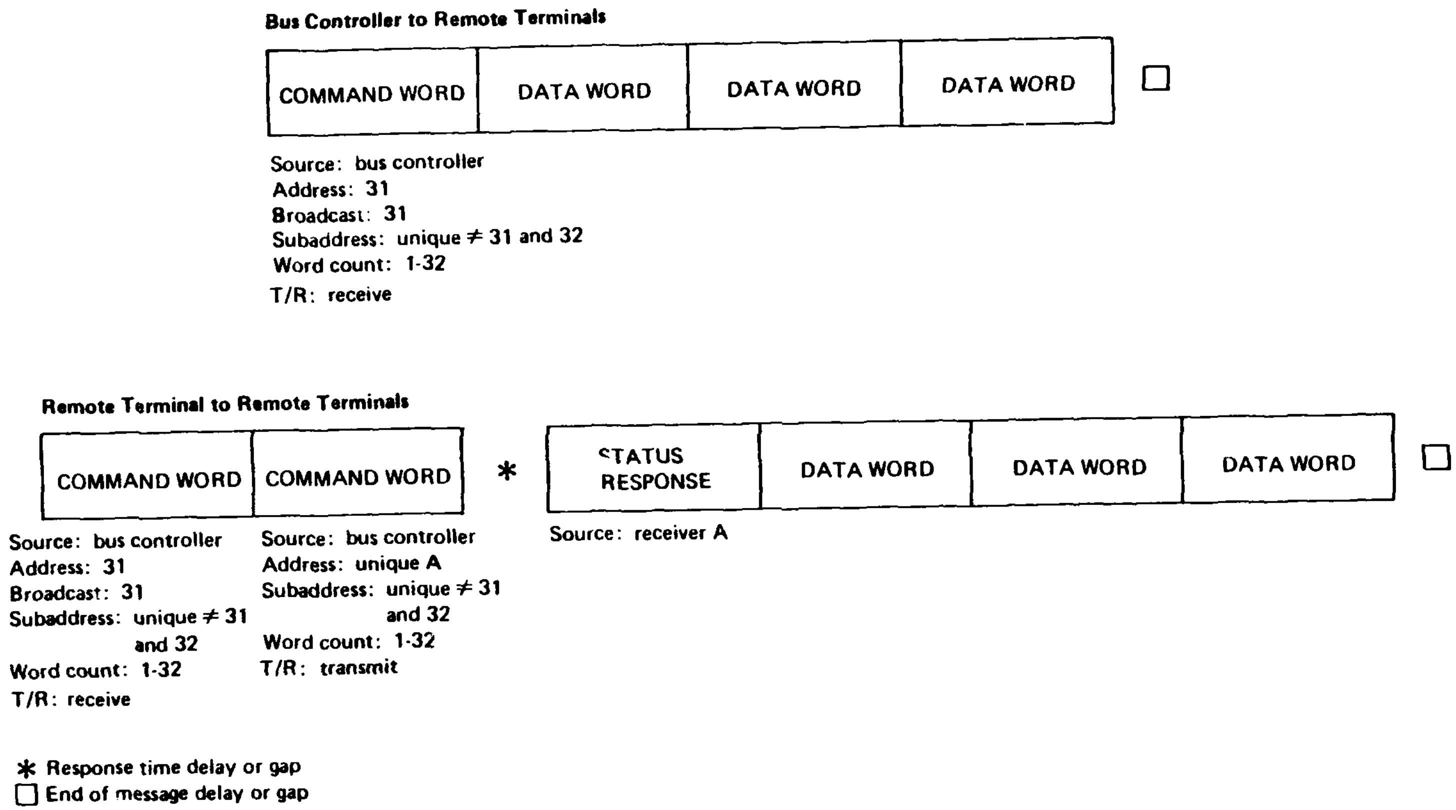


Figure 7-9. Multiple-Receiver Data Message Formats

validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

Mode commands are used to manage the data bus system and are considered a necessary overhead requirement to properly control the data flow. The overhead requirements are provided by command words and status words. These header words to each data transmission are required to maintain system data flow within the multiplex system. Command and status words are associated with both control messages and data messages. Message formats within this protocol can be transmitted to a single receiver or to multiple receivers based upon the command word address for the message.

Mode commands are identified by the subaddress/mode field in the command word being set to 32 (00000) or 31 (11111). All control messages originate with the bus controller and are received by a single receiver or by multiple receivers (broadcast). A terminal address value of 31 (11111) in the command word indicates a broadcast message, and any other terminal address is used to identify unique mode commands to terminals on the bus. The mode command information is in the word count/mode code field of the command word and in the attached data word if allowed by the mode command.

The various legal mode commands without and with data word are illustrated in figure 7-10.

4.3.3.7 Intermessage gap. The bus controller shall provide a minimum gap time of 4.0 microseconds (us) between messages as shown on figure 6 and figure 7. This time period, shown as T on figure 8, is measured at point A of the bus controller as shown on figure 9 or figure 10. The time is measured from the mid-bit zero crossing of the last bit of the preceding message to mid-zero crossing of the next command word sync.

This paragraph in the 1553B expands the requirements of the response time (par. 4.3.1 in 1553A), by adding this intermessage gap paragraph. The purpose was to clearly identify that the bus controller shall not transmit contiguous messages (must have a gap) and that the maximum response time (12 us par. 4.3.3.8) does not apply to gaps between messages. The bus controller may issue messages with a gap time greater than 4 us.

4.3.3.8 Response time. The RT shall respond, in accordance with 4.3.3.6, to a valid command word within the time period of 4.0 to 12.0 us. This time period, shown as T on figure 8, is measured at point A of the RT as shown on figure 9 or figure 10. The time is measured from the mid bit zero crossing of the last word as specified in 4.3.3.6 and as shown on figure 6 and figure 7 to the mid-zero crossing of the status word sync.

This paragraph in 1553B relates to paragraph 4.3.1 in 1553A with the following changes:

a. The maximum response time was increased by 100% (5 to 10 us or 7 to 12 us when using the measurement techniques described below).

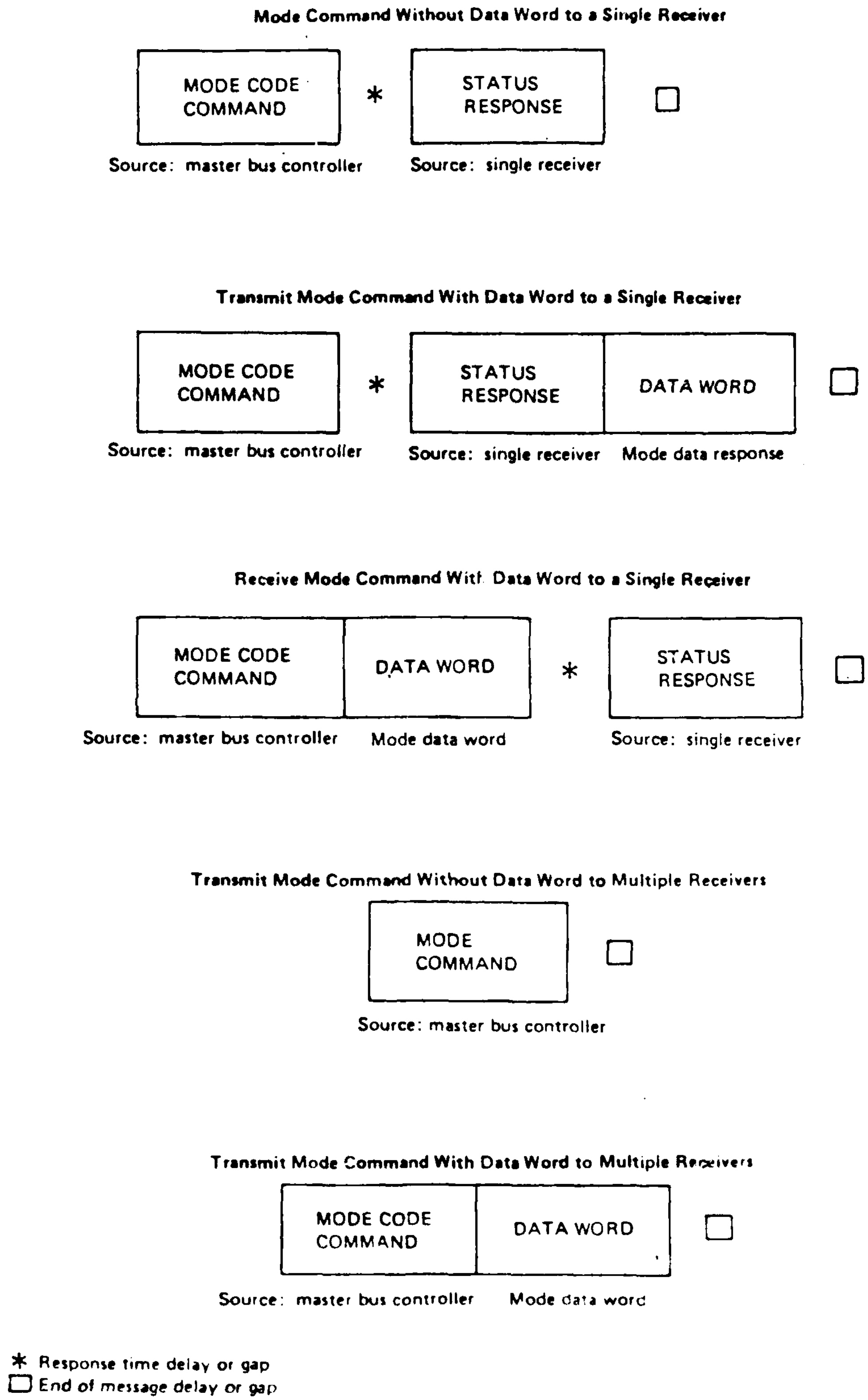


Figure 7-10 Mode Command Transfer Formats

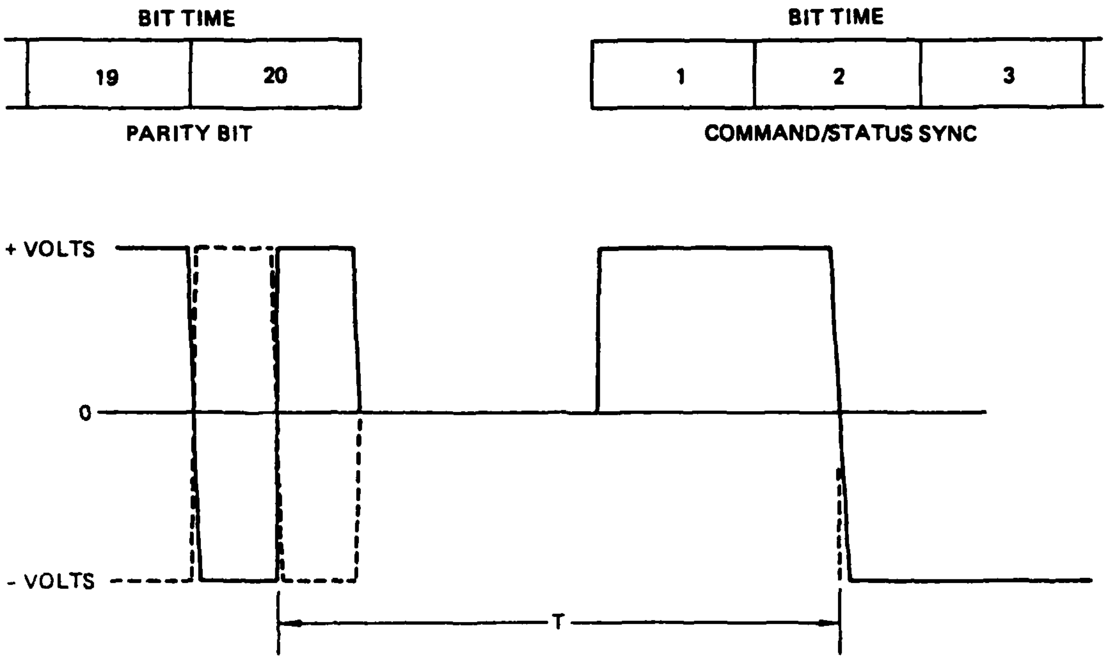


Figure 8 of 1553B. Intermessage Gap and Response Time

b. The point of measurement to establish the time was identified and was chosen to be a different point than was usually interpreted to be in 1553A. The 4 to 12 us response time will allow more hardware design flexibility in the multiplex interface area. Also, the measurement technique was undefined in 1553A and because it is hard to determine when the multiplex line is quiet (dead), the measurement is easier to make if the previous midbit (zero) crossing and next midbit crossing are examined.

> 4.3.3.9 <u>Minimum no-response time-out</u>. The minimum time that a terminal shall wait before considering that a response as specified in 4.3.3.8 has not occurred shall be 14.0 us. The time is measured from the mid-bit zero crossing of the last bit of the last word to the mid-zero crossing of the expected status word sync at Point A of the terminal as shown on figure 9 or figure 10.

This new requirement of 1553B is provided to clarify the minimum time that a bus controller shall wait before concluding that the RT is not going to respond as requested. This is measured from the end of its transmission (last midbit crossing) to the expected response (first midbit crossing). Notice that this represents the minimum wait time on the same bus where the previous message was requested from the RT.

> 4.4 TERMINAL OPERATION
>
> This paragraph in 1553B was provided to clarify the various terminals identified in the standard and their performance

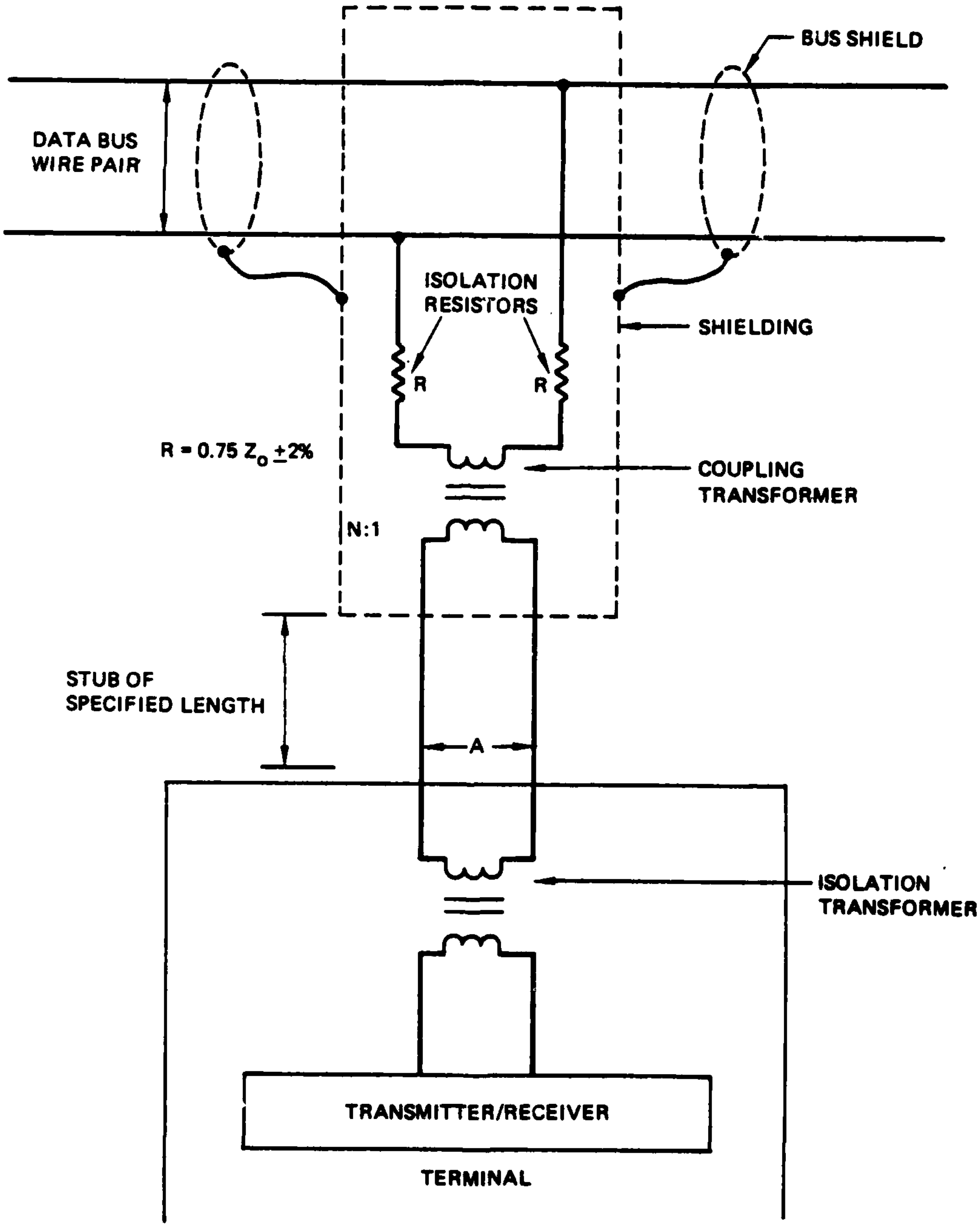


Figure 9 of 1553B. Data Bus Interface Using Transformer Coupling

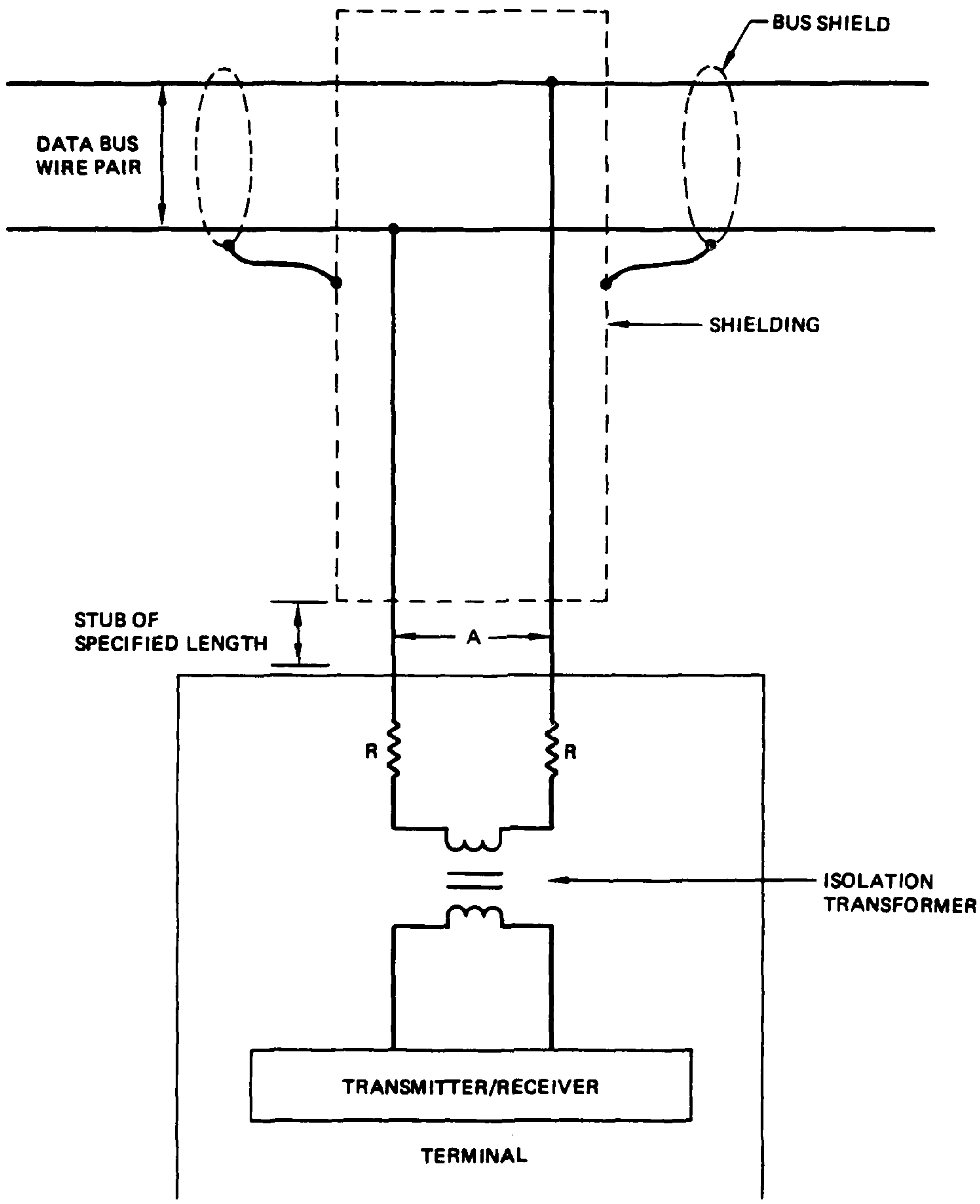


Figure 10 of 1553B. Data Bus Interface Using Direct Coupling

requirements. The first section covers common operational requirements that apply to all devices connected to the data bus system. Specific requirements include bus controller (par. 4.4.2), remote terminal (par. 4.4.3), and bus monitor (par. 4.4.4).

4.4.1 <u>Common operation</u>. Terminals shall have common operating capabilities as specified in the following paragraphs.

4.4.1.1 <u>Word validation</u>. The terminal shall insure that each word conforms to the following minimum criteria:

a. The word begins with a valid sync field.
b. The bits are a valid Manchester II code.
c. The word parity is odd.

When a word fails to conform to the preceding criteria, the word shall be considered invalid.

4.4.1.2 <u>Transmission continuity</u>. The terminal shall verify that the message is contiguous as defined in 4.3.3.6. Improperly timed data syncs shall be considered a message error.

4.4.1.3 <u>Terminal fail-safe</u>. The terminal shall contain a *hardware implemented time-out to preclude a signal* transmission of greater than 800.0 us. This hardware shall not preclude a correct transmission in response to a command. Reset of this time-out function shall be performed by the reception of a valid command on the bus on which the time-out has occurred.

This paragraph describes the common operation associated with terminals connected to the data bus system. The performance requirements include: word validation, transmission continuity, and terminal fail-safe.

The word validation paragraph has been modified to explain more fully the detection and response required. These requirements are contained in 1553B paragraphs (word validation, 4.4.1.1), (transmission continuity, 4.4.1.2), and (invalid data reception, 4.4.3.6). The new and modified paragraphs should provide sufficient information concerning invalid words or invalid messages.

The terminal fail-safe requirement prevents excessive transmissions on a data bus by a single transmitter, which would preclude its effective use. Changes in 1553B to 800 us (par. 4.4.1.3) instead of 660 us (par. 4.3.2) will allow less accurate analog or relaxed digital timers with more independence of the timer circuits to be used in the current design. In addition, there were several mechanisms to reset this fail-safe timer described in 1553B. These include the following:

a. Reset of this timeout function shall be performed by the reception of a valid command on the bus on which the timeout has occurred (par. 4.4.1.3).

b. The mode command override transmitter shutdown (00101 for two-bus system or 10101 for multiple-bus system) on an alternative bus can also be used to reset the timer.

c. The mode command reset remote terminal (01000) causes the remote terminal to assume a power-up initialized state that can also be used to reset the timer.

Both b and c are optional ways to reset the timer and may depend on the system and hardware implementations. However, the preferred reset approach is to transmit the appropriate mode code.

> 4.4.2 Bus controller operation. A terminal operating as a bus controller shall be responsible for sending data bus commands, participating in data transfers, receiving status responses, and monitoring system status as defined in this standard. The bus controller function may be embodied as either a stand-alone terminal, whose sole function is to control the data bus(s), or contained within a subsystem. Only one terminal shall be in active control of a data bus at any one time.

This paragraph is generally the same paragraph in both 1553A (4.5) and 1553B.

> 4.4.3 Remote Terminal
>
> 4.4.3.1 Operation. A remote terminal (RT) shall operate in response to valid commands received from the bus controller. The RT shall accept a command word as valid when the command word meets the criteria of 4.4.1.1, and the command word contains a terminal address which matches the RT address or an address of 11111, if the RT has the broadcast option.

The remote terminal operation has been expanded in 1553B to include the broadcast option and additional definitions associated with: superseding valid commands (par. 4.4.3.2), invalid commands (par. 4.4.3.3), illegal commands (par. 4.4.3.4), valid data reception (par. 4.4.3.6), and invalid data reception (par. 4.4.3.6).

> 4.4.3.2 Superseding valid commands. The RT shall be capable of receiving a command word on the data bus after the minimum intermessage gap time as specified in 4.3.3.7 has been exceeded, when the RT is not in the time period T as specified in 4.3.3.8 prior to the transmission of a status word, and when it is not transmitting on that data bus. A second valid command word sent to an RT shall take precedence over the previous command. The RT shall respond to the second valid command as specified in 4.3.3.8.

The superseding valid command requirement clarifies the gap time issue in 1553A (par. 4.3). "A second valid command word sent to a terminal after it is already operating on one shall invalidate the first command and cause the RT to begin operation on the second command." This phrase in 1553A can be misinterpreted to indicate that near back-to-back (no gap or only 4 us gap)

commands can be sent to a terminal on the same bus and the terminal will respond to the second command.

However, this was not the intention because in certain cases an RT responding with a status word slightly greater than 4 us would collide with the second command being transmitted by the bus controller. The intended purpose for this requirement is to allow the bus controller to reissue an identical transmission or issue a new transmission on the same bus to the same RT, when an RT fails to respond to a command on that bus. This method is described in the 1553B by requiring a minimum time T (greater than a gap time but less than a full 32-word message) to occur prior to transmitting the second command. Therefore, the bus controller is assured that the RT is not responding and a new command on the same bus is appropriate. Figure 8 in 1553B demonstrates this intermessage gap problem and solution.

> 4.4.3.3 Invalid commands. A remote terminal shall not respond to a command word which fails to meet the criteria specified in 4.4.3.1.

Command words that fail to meet the word validation requirements cause the system to continue to "look" for a valid command word. When this condition occurs, no change occurs to the status word and no response is transmitted by the RT. This operation is identified as invalid command. This paragraph is used to cover failure in the decoding process of the command word. To prevent multiple responses by two or more terminals to a command word (one without a failure and one with) the terminal that cannot absolutely validate a command word must take the safe approach and reset the circuitry and continue to look for a valid command word that meets its particular requirements (address). All RT's should use this approach of not responding, when there is a question about the commands. This approach is considered to be a fail-passive approach providing the least impact on the multiplex system.

> 4.4.3.4 Illegal command. An illegal command is a valid command as specified in 4.4.3.1, where the bits in the subaddress/mode field, data word count/mode code field, and the T/R bit indicate a mode command, subaddress, or word count that has not been implemented in the RT. It is the responsibility of the bus controller to assure that no illegal commands are sent out. The RT designer has the option of monitoring for illegal commands. If an RT that is designed with this option detects an illegal command and the proper number of contiguous valid data words as specified by the illegal command word, it shall respond with a status word only, setting the message error bit, and not use the information received.

Illegal commands are command words that have passed the word validation test but do not comply with the system's capability. These include command words where the subaddress-mode field, data word/mode code field, or the T/R bit are set so that they represent conditions not allowed in the system. These include both conditions not allowed by the standard and any additional condition not allowed in a particular system design. The responsibility for not allowing illegal commands to be transmitted is given to the bus

controllers. Since the bus controller is responsible for all command/response message communications, it will be a design goal that the bus controller not transmit an invalid command.

Two methods can be provided to meet this requirement: (1) careful generation of bus controller commands in the development of the system and tight control of the change process during operational use and (2) examination of failure modes of the controller hardware and software to determine potentially illegal command generations and transmissions. An additional method of rejecting illegal commands in the multiplex system can only be provided by circuitry within the receiving remote terminal. This approach is an optional capability for remote terminals built to the 1553B standard. If an RT with this capability detects an illegal command that meets all other validation requirements, the RT shall respond with a status word with only the message error bit set and not use the information sent or disregard the request for information.

> 4.4.3.5 Valid data reception. The remote terminal shall respond with a status word when a valid command word and the proper number of contiguous valid data words are received, or a single valid word associated with a mode code is received. Each data word shall meet the criteria specified in 4.4.1.1.

The purpose of the valid data reception in 1553B was to clearly state when a message containing at least one data word would be responded to by the appropriate RT. Previous systems have taken different approaches to messages with various failures (e.g., under word count, over word count, parity errors in words, gaps in word transmissions). Therefore, this requirement was established to identify the only time status words would be transmitted by the RT after the reception of a data message with at least one data word. It should be noted that one other message format will produce a status word response: mode code without data word transmitted to a specific RT (not broadcast).

> 4.4.3.6 Invalid data reception. Any data word(s) associated with a valid receive command that does not meet the criteria specified in 4.4.1.1 and 4.4.1.2 or an error in the data word count shall cause the remote terminal to set the message error bit in the status word to a logic one and suppress the transmission of the status word. If a message error has occurred, then the entire message shall be considered invalid.

In contrast to the valid data reception status response, certain action is required when an invalid data reception occurs. This paragraph in 1553B is an expanded version of the two 1553A paragraphs 4.2.5.4.4 and 4.2.3.5.3.3. This again assumes message formats with associated data word(s), thus mode code commands without a data word are rightly excluded from this group; however, in contrast to valid data reception where broadcast message protocol were excluded, here they are included. Therefore, all message formats containing at least one data word (e.g., broadcast data messages, nonbroadcast data messages, broadcast mode codes with a data word, and mode codes with a data word) are included in this requirement. As stated in the requirement, the message command word has been validated and the error occurs in the data word portion of the message. The withholding or suppression of the status response alerts the bus controller error detection

electronics to the fact that an incomplete message has occurred and some level of error recovery must occur. The setting of the message error bit in the status that remains in the RT will provide additional information to the error recovery circuitry only if the bus controllers request the status word using the appropriate mode code.

Also notice that the requirement is that the entire received message be considered invalid. This message invalidation requirement may cause some systems like electrical multiplex (EMUX) a problem. Since the EMUX system usually have bit-oriented data rather than word or multiple words (message) oriented data, errors in a word following the reception of good data will invalidate good data. It has been proposed that such a system invalidate all data words from the failure to the end of the message and use previously good data words. This approach, however, has not been allowed. Regardless of the approach, some system mechanisms will store the data and then tag the message as being invalid; others will not allow the user to receive the data. In the first case, it is the responsibility of the user to examine the message valid indication prior to using the data; however, in the second case, the user must recognize that the data has not been updated.

> 4.4.4 Bus monitor operation. A terminal operating as a bus monitor shall receive bus traffic and extract selected information. While operating as a bus monitor, the terminal shall not respond to any message except one containing its own unique address if one is assigned. All information obtained while acting as a bus monitor shall be strictly used for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller.

A terminal may operate in the bus monitor mode for two reasons: (1) information recording for offline analysis and (2) information source for backup bus controller. The unique feature of this mode is that it has the ability to decode and accept for data storage any or all messages transmitted on the data bus without the knowledge of or without affecting the operation of multiplex system or the terminal(s) attached to the bus. It also has the option of not being addressable as a terminal attached to the bus. If this is the case, it acts in the "listen capability only" to the system. In this implementation, data cannot be sent to it specifically, but the monitor may collect data by recording message traffic. However, the same terminal may operate in the remote terminal and potential bus controller (backup bus controller) mode as well as having the additional capability to monitor and store all message traffic or an internally derived subset of all messages. It is because of this second capability, its nonpassive nature, that a terminal with the monitor mode is an extremely powerful device in the multiplex system.

4.5 Hardware Characteristics

The following discussion will provide a summary and comparison of MIL-STD-1553A/B requirements that have significant effect on the hardware characteristics (1553, par. 4.5). A detailed comparison of these subparagraphs in 1553A and 1553B is provided in tables 7-2 and 7-3.

Table 7-2. Comparison of Data Bus Characteristics

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
1. Twisted, shielded, jacketed	Yes 4.2.4.1	Yes 4.5.1.1
2. Minimum cable shield coverage	80% 4.2.4.1	75% 4.5.1.1
3. Minimum cable twist	1 twist/in (12 twists/ft) 4.2.4.1	4 twists/ft 4.5.1.1
4. Wire-to-wire distributed capacitance (maximum)	30 pF/ft 4.2.4.1	30 pF/ft 4.5.1.1
5. Characteristic impedance of cable	70 ± 10% at 1 MHz 4.2.4.2	Nominal 70 to 80 at 1 MHz 4.5.1.2
6. Cable attenuation	1 dB/100 ft at 1 MHz 4.2.4.3	1.5 dB/100 ft at 1 MHz 4.5.1.3
7. Cable length	300 ft maximum 4.2.4.4	Unspecified —
8. Cable termination using a resistance at both ends	Characteristic impedance 4.2.4.6	Nominal characteristic impedance ±2% 4.5.1.4
9. Cable stubbing	Transformer coupling for stubs longer than 1 ft but less than 20 ft; direct coupling if stub is less than 1 ft; maximum stub length of 20 ft 4.2.4.5 Figure 7	Transformer coupling or direct coupling allowed; maximum stub length suggested 20 ft 4.5.1.5.1 or 4.5.1.5.2 Figures 9 or 10
10. Cable coupling (connector)	Compatible with Amphenol type 31-235 or Trompeter type TEI-14949-E137 receptacles and Amphenol type 31-224 or Trompeter type TEI-14949-PL36 plugs 4.2.4.6	Unspecified —

Table 7-2. Comparison of Data Bus Characteristics (Continued)

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
11. Cable coupling shielded box	Shielded coupler box 4.2.4.6	75% coverage, minimum 4.5.1.5.1.3 and 4.5.1.5.2.2
12. Coupling transformer turns ratio	Unspecified –	1:1.41 ± 3% higher turns on isolation resistor side of stub 4.5.1.5.1.1
13. Transformer open circuit impedance	Unspecified –	3,000 ohms over frequency of 75 kHz -1 MHz with 1V RMS sine wave 4.5.1.5.1.1.1
14. Transformer waveform integrity	Unspecified –	Droop not to exceed 20% overshoot and ringing less than ± 1V peak undertest of figure 11 4.5.1.5.1.1.2
15. Transformer common mode rejection	Unspecified –	45 dB at 1 MHz 4.5.1.5.1.1.3
16. Fault isolation– Isolation resistor in series with data bus cable (coupler)	$R = 0.75\ Z_o{}^{*} \pm 5\%$ 4.2.5.2 Figure 7	$R = 0.75\ Z_o{}^{*} \pm 2\%$ 4.5.1.5.1.2
Direct coupled case with the isolation resistor in the RT		$R = 55$ ohms ± 2% 4.5.1.5.2.1 Figure 10
17. Impedance across the data bus for any failure of coupling transformer, cable stub, or terminal receiver and transmitter transformer coupling	No less than $1.5\ Z_o{}^{*}$ 4.2.5.2	No less than $1.5\ Z_o{}^{*}$ 4.5.1.5.1.2
Direct coupling		No less than 110 ohms 4.5.1.5.2.3
18. Stub voltage requirements and input level transformer coupling	**Range of the 0.5V to 10V peak; 1.0V to 20V p-p, l-l 4.2.5.4.1 Figure 7	**Range of 1.0V to 14.0V p-p**, l-l with one fault as stated in 17 above 4.5.1.5.1.4 Figure 9

*Z_o = cable nominal characteristic impedance

**Assumes one fault of a coupling transformer, cable stub, or terminal receiver or transmitter

Table 7-2. Comparison of Data Bus Characteristics (Concluded)

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
Direct coupling	**Range of the 0.5V to 10V peak; 1.0V to 20V p-p, l-l 4.2.5.4.1 Figure 7	**Range of 1.4V to 20V p-p, l-l with one fault as stated in 17 above 4.5.1.5.2.3 Figure 10
19. Wiring and cabling for electromagnetic capability	MIL-E-6051 4.2.4.7 MIL-STD-1553A	MIL-E-6051 4.5.1.5.3 MIL-STD-1553B

**Assumes one fault of a coupling transformer, cable stub, or terminal receiver or transmitter

Table 7-3. Comparison of Terminal Characteristics

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
1. Output level—transformer coupling	±3.0V to ±10V peak (6.0V to 20.0V p-p) I-I with no faults; with one fault of a coupling transformer, cable stub, or terminal receiver-transmitter, ±2.25V to ±11.25V peak (4.5V to 15V p-p) I-I 4.2.5.3.1	With RL = 70 ± 2%, 18.0V to 27.0V p-p, I-I 4.5.2.1.1.1 Figure 12
Direct coupling		With RL = 35 ± 2%, 6.0V to 9.0V p-p, I-I 4.5.2.2.1.1 Figure 12
2. Output waveform—Zero crossing deviation	±25 ns 4.2.5.3.2 Point C, figures 7 and 8	±25 ns 4.5.2.1.1.2 Figure 12
Rise and fall time (10% to 90%)	≥100 ns 4.2.5.3.2 Figure 8	100 to 300 ns 4.5.2.1.1.2 Figure 13
Transformer coupling distortion (including overshoot and ringing)	Unspecified	±900-mV peak, I-I 4.5.2.1.1.2 Point A, figure 12
Direct coupling distortion (including overshoot and ringing)	Unspecified	±300-mV peak, I-I 4.5.2.2.1.2 Point A, figure 12
3. Output noise—Transformer coupling	10-mV p-p, I-I 4.2.5.3.3 Point A, figure 7	14-mV, RMS, I-I 4.5.2.1.1.3 Point A, figure 12

Table 7-3. Comparison of Terminal Characteristics (Continued)

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
Direct coupling		5-mV, RMS, l-l 4.5.2.2.1.3 Point A, figure 12
4. Output symmetry (after 2.5 μs of midbit crossing of the last parity bit— Transformer coupling	Unspecified	±250-mV peak, l-l 4.5.2.1.1.4 Point A, figure 12
Direct coupling	Unspecified	±90-mV peak, l-l 4.5.2.2.1.4 Point A, figure 12
5. Input waveform— Maximum zero crossing deviation	Unspecified	±150 ns 4.5.2.1.2.1 Point A, figures 9 or 10
6. Input signal response range Transformer coupling	±0.5V to ±10.0V peak (1.0V to 20V p-p), l-l 4.2.5.4.1 Point C, figure 7	0.86V to 14.0V p-p, l-l 4.5.2.1.2.1 Point A, figure 9
Direct coupling		1.2V to 20V p-p, l-l 4.5.2.2.2.1 Point A, figure 10
7. Input signal no response range Transformer coupling	Unspecified	0.0V to 0.2V p-p, l-l 4.5.2.1.2.1 Point A, figure 9
Direct coupling	Unspecified	0.0V to 0.28V p-p, l-l 4.5.2.2.2.1 Point A, figure 10

Table 7-3. Comparison of Terminal Characteristics (Concluded)

Requirement	Requirement and MIL-STD-1553A paragraph	Requirement and MIL-STD-1553B paragraph
8. Common mode rejection	±10.0V peak, line-to-ground, dc to 2 MHz 4.2.5.4.2 Point A, figure 7	±10.0V peak, line-to-ground, dc to 2 MHz 4.5.2.1.2.2 or 4.5.2.2.2.2 Point A, *figures* 9 or 10
9. Input impedance— Transformer coupling	Minimum of 2,000 ohms over a frequency range of 100 kHz to 1 MHz, l-l 4.2.5.4.3 Point C, figure 7	Minimum of 1,000 ohms over a frequency range of 75 kHz to 1 MHz, l-l 4.5.2.1.2.3 Point A, figure 9
Direct coupling		Minimum of 2,000 ohms over a frequency range of 75 kHz to 1 MHz, l-l 4.5.2.2.2.3 Point A, figure 10
10. Noise rejection or error rate— Transformer coupling	Maximum bit error rate of 10^{-12} and a maximum incomplete message rate of 10^{-6} in a configuration of one bus controller on a 20-ft stub with a minimum of 100 ft of main bus cable between coupling boxes; test is conducted *in presence of magnetic field per* MIL-STD-462 method RS02 (spike test) with the limits of MIL-STD-461 RS02 4.3.3	Maximum of one part in 10^7 word error in the presence of additive white gaussian noise of 140-mV RMS over a bandwidth of 1.0 kHz to 4 MHz; input voltage 2.1V p-p, l-l Point A, *figure* 9, *accept/reject* Table *II* 4.5.2.1.2.4
Direct coupling		Maximum of one part in 10^7 word error rate in the presence of additive white gaussian noise of 200-mV RMS over a bandwidth of 1.0 kHz to 4 MHz; input voltage 3.0V p-p, l-l Point A, figure 10, accept/reject Table II 4.5.2.2.2.4

The hardware characteristic section of 1553B examines data bus characteristics (par. 4.5.1) and terminal characteristics (par. 4.5.2). This section is similar to the terminal operation paragraphs (4.3) and the transmission line (4.2.4) of 1553A. Paragraph 4.4 in 1553A (Terminal to Subsystem Interface) has been deleted from 1553B completely. This deletion was consistent with the emphasis on a data bus protocol standard and an electrical multiplex interface requirement treating the terminals interfacing to the multiplex bus as black box interfaces and not defining any internal interfaces.

4.5.1 Data Bus Characteristics

4.5.1.1 Cable. The cable used for the main bus and all stubs shall be a two conductor, twisted, shielded, jacketed cable. The wire-to-wire distributed capacitance shall not exceed 30.0 picofarads per foot. The cables shall be formed with not less than four twists per foot where a twist is defined as a 360 degree rotation of the wire pairs; and, the cable shield shall provide a minimum of 75.0 percent coverage.

4.5.1.2 Characteristic impedance. The nominal characteristic impedance of the cable (Z_0) shall be within the range of 70.0 Ohms to 85.0 Ohms at a sinusoidal frequency of 1.0 megahertz (MHz).

4.5.1.3 Cable attenuation. At the frequency of 4.5.1.2, the cable power loss shall not exceed 1.5 decibels (dB)/100 feet (ft).

Table 7-4 contains a summary listing of the data bus and coupling requirements contained in 1553A and 1553B. The characteristics of the twisted shielded pair cable have been relaxed to allow selection of cable types from a variety of manufacturers. It has been shown that minor variations from the specified cable characteristics do not significantly affect the system performance.

A great deal of concern and confusion has resulted because of the cable network requirements, including bus length, coupling, and stubbing. 1553 and 1553A did not provide adequate guidelines for bus network design, especially for the transformer coupled stub. 1553A defined a maximum cable length of 300 ft for the main bus while 1553B chose not to specify a maximum main bus length since it is reasoned that the cable length, number of terminals, and length of stubs are all subject to trade-off and must be considered in the design for reliable system operation. In other words, an arbitrary limit of 300 ft should not be applied since all parameters of the network must be considered.

4.5.1.4 Cable termination. The two ends of the cable shall be terminated with a resistance, equal to the selected cable nominal characteristic impedance (Z_0) ± 2.0 percent.

4.5.1.5 Cable stub requirements. The cable shall be coupled to the terminal as shown on figure 9 or figure 10. The use of long stubs is discouraged, and the length of a stub should be minimized. However, if installation requirements dictate,

Table 7-4. Summary of Data Bus and Coupling Requirements

Parameter	MIL-STD-1553A	MIL-STD-1553B
Transmission line		
• Cable type	Twisted-shielded pair	Twisted-shielded pair
• Capacitance (wire-to-wire)	30 pF/ft, maximum	30 pF/ft, maximum
• Twist	1/in, minimum	4/ft (0.33/in), minimum
• Characteristic impedance (Z_o)	70 ohms ± 10% at 1.0 MHz	70 to 85 ohms at 1.0 MHz
• Attenuation	1.0 dB/100 ft at 1.0 MHz, maximum	1.5 dB/100 ft at 1.0 MHz, maximum
• Length of main bus	300 ft, maximum	Not specified
• Termination	Two ends terminated in resistors equal to Z_o	Two ends terminated in resistors equal to Z_o ± 2%
• Shielding	80% coverage, minimum	75% coverage, minimum

Table 7-4. Summary of Data Bus and Coupling Requirements (Continued)

Parameter	MIL-STD-1553A	MIL-STD-1553B
Cable coupling		
• Stub definition	Short stub <1 ft Long stub >1 to 20 ft (20 ft, *maximum*)	Short stub <1 ft Long stub >1 to 20 ft (may be exceeded)
• Coupler requirement	All connections use external coupler box; connectors specified (ref. fig. 4.2-1)	Direct coupled, short stub transformer coupled, long stub (ref. fig. 4.2-2)
• Coupler transformer		
• Turns ratio	Not specified	1 to 1.41
• Input impedance	Not specified	3,000 ohms, minimum (75.0 kHz to 1.0 MHz)
• Droop	Not specified	20% maximum (250 kHz)
• Overshoot and ringing	Not specified	±1.0V peak (250-kHz square wave with 100-ns maximum rise and fall time)
• Common mode rejection	Not specified	45.0 dB at 1.0 MHz
• Fault protection	Resistor in series with each connection equal to $(0.75Z_0) \pm 5\%$ ohms	Resistor in series with each connection equal to $(0.75Z_0) \pm 2.0\%$ ohms
• Stub voltage	±0.5V to ±10.0V, peak, l-l (1.0V to 20.0V, p-p, l-l); nominal signal level for terminal response	1.0V to 14.0V p-p, l-l, minimum signal voltage (transformer coupled) 1.4V to 20.0V, p-p, l-l, minimum signal voltage (direct coupled)

stub lengths exceeding those lengths specified in 4.5.1.5.1 and 4.5.1.5.2 are permissible.

4.5.1.5.1 Transformer coupled stubs. The length of a transformer coupled stub should not exceed 20 feet. If a transformer coupled stub is used, then the following shall apply.

4.5.1.5.1.1 Coupling transformer. A coupling transformer, as shown on figure 9, shall be required. This transformer shall have a turns ratio of 1:1.41 $\pm$3.0 percent, with the higher turns on the isolation resistor side of the stub.

A generalized multiplex bus network configuration is shown in figure 1 of 1553B. The main bus is terminated at each end in the cable characteristic impedance to minimize reflections caused by transmission line mismatch. With no stubs attached, the main bus looks like an infinite length transmission line and therefore there are no disturbing reflections. When the stubs are added for connection of the terminals, the bus is loaded locally and a mismatch occurs with resulting reflections. The degree of mismatch and signal distortions caused by reflections are a function of the impedance (Z) presented by the stub and terminal input impedance. In order to minimize signal distortion, it is desirable that the stub maintain a high impedance. This impedance is reflected back to the main bus. At the same time the impedance needs to be kept low so that adequate signal power will be delivered to the receiver input. Therefore, a trade-off and compromise between these conflicting requirements is necessary to achieve the specified signal-to-noise ratio and system error rate performance. Two methods for coupling a terminal to the main bus are defined in 1553B, transformer coupling and direct coupling. The two methods are shown in figures 9 and 10 of 1553B. Transformer coupling is usually used with long stubs (1 to 20 ft) and requires a coupler box, separate from the terminal, located near the junction of the main bus and stub. Direct coupling is usually limited for use with stubs of less than 1 ft. Fault isolation resistors (R) are included to provide protection for the main bus in case of a short circuit in the stub or terminal. The coupler-transformer characteristics defined in 1553B and listed in table 7-4 provide a compromise between the signal level and distortion characteristics delivered to the terminals. The coupler transformer turns ratio (1:1.41) provides impedance transformation for both terminal reception and transmission. The improvement of stub load impedance is a result of impedance transformation, which is proportional to the square of the turns ratio, assuming an ideal coupler transformer.

As indicated above, the 1:1.41 transformer also provides ideal termination of the stub for transmission of signals from the terminal to the main bus. The impedance at main bus is

$$Z_B = \frac{Z_0}{2} + 2R \qquad (1)$$

where

$R = 0.75\ Zo$

$$Z_B = 0.5Zo + 1.5Zo = 2Zo \text{ ohms} \qquad (2)$$

The reflected impedance, Z_R, from the bus to the stub due to the transformer impedance transformation is

$$Z_R = \frac{Z_B}{1.41^2} = \frac{2Zo}{2} = Zo \tag{3}$$

Therefore, the coupler transformer specified in 1553B provides the characteristics desired for reducing reflections and maintaining signal levels for systems where long stubs are required.

> 4.5.1.5.1.1.1 <u>Transformer input impedance</u>. The open circuit impedance as seen at point B on figure 11 shall be greater than 3000 Ohms over the frequency range of 75.0 kilohertz (KHz) to 1.0 megahertz (MHz), when measured with a 1.0V root-mean-square (RMS) sin wave.

The transformer open circuit impedance (Zoc) is required to be greater than 3K ohms in 1553B systems. The measurement is made looking into the higher turns winding (1.41) with a 75 kHz to 1 MHz sine wave signal. The test amplitude at the transformer winding is adjusted to 1V rms. The critical factors in achieving the 3K ohm Zoc is the distributed capacitance of the windings and the transformer primary inductance. The inductance of the transformer must be large enough to provide the open circuit impedance at 75 kHz, and the distributed capacitance should be small enough to maintain the open circuit impedance at the 1 MHz test frequency. The inductance may obviously be increased by increasing the number of turns on the transformer. This technique, however, tends to increase the distributed capacitance, degrading high frequency-performance and therefore causing waveform integrity and common mode rejection to suffer.

> 4.5.1.5.1.1.2 <u>Transformer waveform integrity</u>. The droop of the transformer using the test configuration shown on figure 11 at point B, shall not exceed 20.0 percent. Overshoot and ringing as measured at point B shall be less that ± 1.0 V peak. For this test, R shall equal 360.0 ohms ± 5.0 percent and the input A of figure 11 shall be a 250.0 KHz square wave, 27.0 V peak-to-peak, with a rise and fall time no greater than 100 nanoseconds (ns).

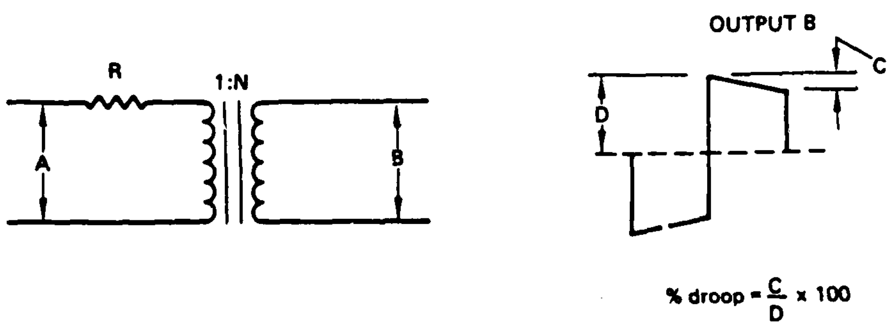


Figure 11 of 1553B. Coupling Transformer

The ability of the coupler transformer to provide a satisfactory signal is specified in the droop, overshoot, and ringing requirements of 1553B shown in figure 7-11. Droop is specified at 20% maximum when driving the transformer with a 250 kHz, 27V p-p square wave. The test for the droop characteristic is made by driving the low turns winding through a 360 ohm resistor and measuring the signal at the open-circuited high side winding. The droop of the transformer is determined mainly by the primary inductance. Since the primary inductance also provides the 3K ohm open circuit impedance, the inductance should be made as high as possible without degrading the high-frequency performance of the transformer. Ringing and overshoot on the transformer signal is also shown in figures 7-11. The ±1V limit on these high-frequency perturbations can be achieved through careful attention to leakage inductance and transformer capacitance.

4.5.1.5.1.1.3 Transformer common mode rejection. The coupling transformer shall have a common mode rejection ratio greater than 45.0 dB at 1.0 MHz.

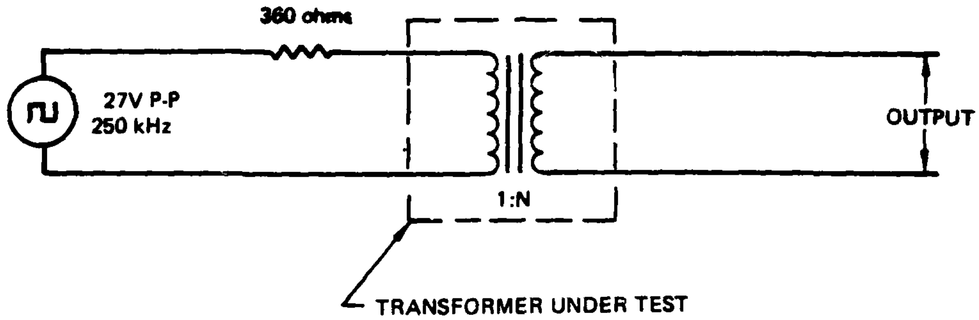
The common mode rejection of the isolation transformer is required to be greater than 45.0 dB. The common mode test shown in figure 7-12 consists of driving the high turns winding while measuring the differential signal across the high side. Common mode rejection can be improved by minimizing the interwinding capacitance and the core-to-winding capacitance.

4.5.1.5.1.2 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 0.75 Z_0 minus 2.0 percent, where Z_0 is the selected cable nominal characteristic impedance. The impedance placed across the data bus cable shall be no less than 1.5 Z_0 failure of the coupling transformer, cable stub, or terminal transmitter/receiver.

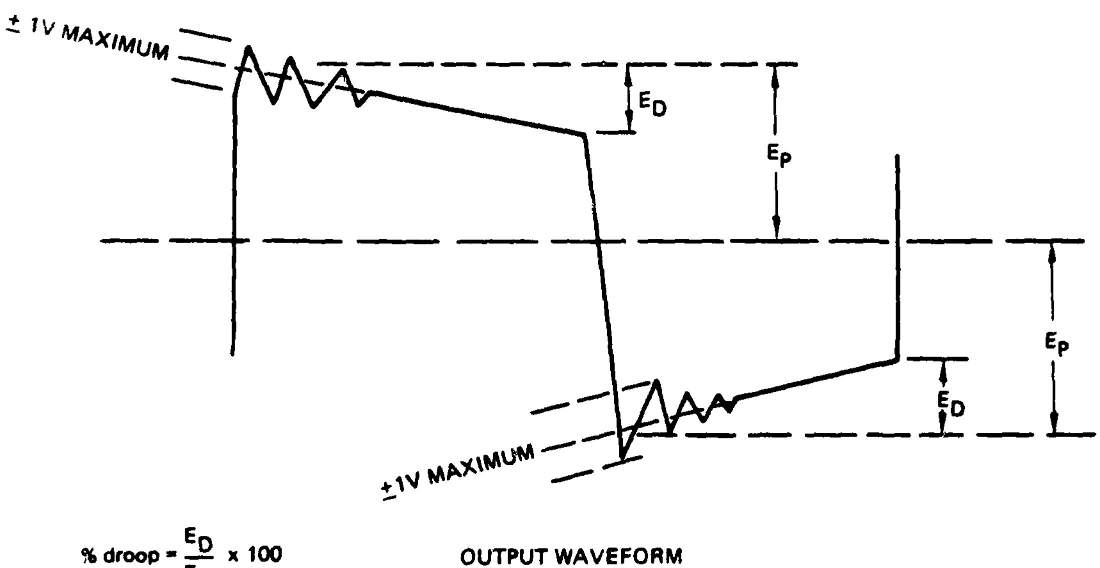
4.5.1.5.1.3 Cable coupling. All coupling transformers and isolation resistors, as specified in 4.5.1.5.1.1 and 4.5.1.4.1.2, shall have continuous shielding which will provide a minimum of 75 percent coverage. The isolation resistors and coupling transformers shall be placed at minimum possible distance from the junction of the stub to the main bus.

4.5.1.5.1.4 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 9 shall have a peak-to-peak amplitude, line-to-line within the range of 1.0 and 14.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault impedance specified in 4.5.1.5.1.2 on the data bus. This shall also include the worst case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.2 Direct coupled stubs. The length of a direct coupled stub should not exceed 1 foot. Refer to 10.5 for comments concerning direct coupled stubs. If a direct coupled stub is used, then the following shall apply.



N = turns ratio (1.41)



$$\% \text{ droop} = \frac{E_D}{E_P} \times 100$$

Figure 7-11. Waveform Test

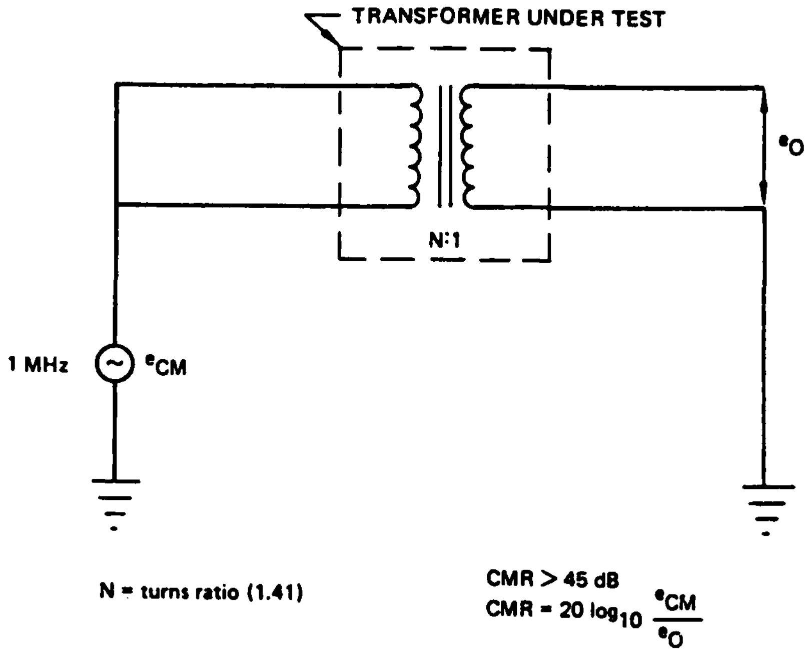


Figure 7-12. Common Mode Test

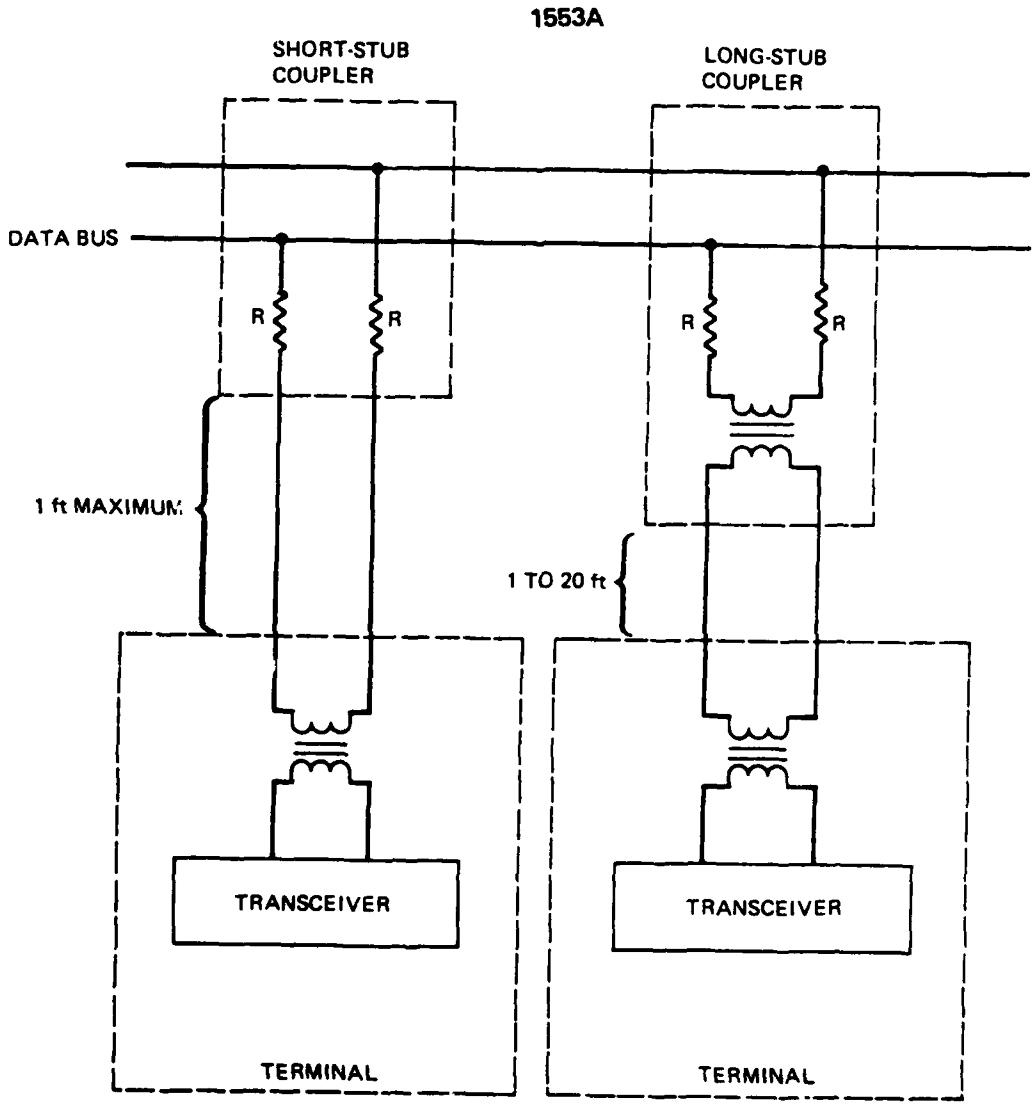
4.5.1.5.2.1 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 55.0 Ohms plus or minus 2.0 percent. The isolation resistors shall be placed within the RT as shown on figure 10.

4.5.1.5.2.2 Cable coupling. All bus-stub junctions shall have continuous shielding which will provide a minimum of 75 percent coverage.

4.5.1.5.2.3 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 10 shall have a peak-to-peak amplitude, line-to-line within the range of 1.4 and 20.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault inpedance of 110 Ohms on the data bus. This shall also include the worst case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.3 Wiring and cabling for EMC. For purposes of electsomagnetic capability (EMC), the wiring and cabling provisions of MIL-E-6051 shall apply.

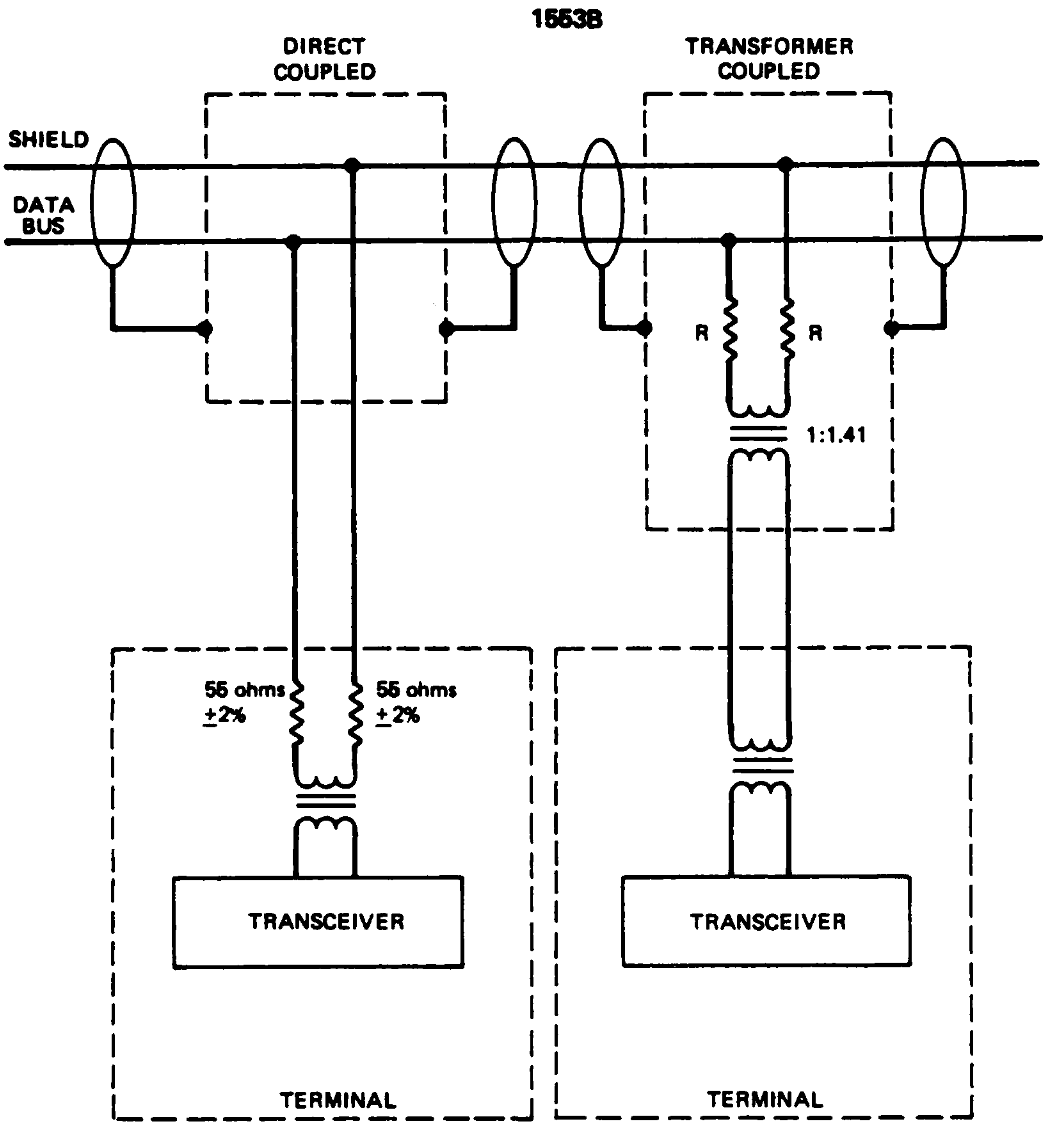
The requirements for couplers specified in 1553A have been modified for 1553B and a comparison of the two requirements is shown in figure 7-13. The major differences in the two requirements are the placement of the isolation



- Isolation resistors: $R = 0.75\ Z^*_o \pm 5\%$
- Isolation transformer: (not specified)

*Nominal characteristic impedance of bus cable: $Z_o = 70 \pm 10\%$ at 1 MHz

Figure 7-13. Coupler Characteristics



- Isolation resistors: $R = 0.75\ Z_o^* \pm 2\%$
- Isolation transformer: turns ratio 1:1.41 ±3%
  (1–terminal winding)
  (1.41–bus winding)

$Z_{oc} > 3K$ at 75 kHz to 1 MHz
1V rms sine wave

Droop: < 20%
Overshoot/ringing: < ±1V
CMR: > 45 dB at 1 MHz
} at 27 V P-P 250 kHz square wave

*Nominal characteristic impedance of bus cable: Z_o = 70 to 85 at 1 MHz

Figure 7-13. Coupler Characteristics (Continued)

resistors for the direct-coupled (short-stub) connection and the characterization of the coupling transformer in the long-stub (transformer-coupled) connection. With the isolation resistors located in the terminal for the direct-coupled case, the need for a separate coupler box is eliminated as long as a reliable shielded splice can be maintained.

The terminal input and output specifications for the transformer-coupled and direct-coupled connections are required to be separated in 1553B because of the effects on signal levels and impedances of the transformer turns ratio being specified as 1:1.41 instead of the assumed 1:1 in 1553A.

The transformer in the 1553B coupler has the turns ratio of 1:1.41. This ratio, together with the 0.75Z fault isolation resistor provides the correct characteristic impedance for terminating the stub:

$$Z\ stub = \left(\frac{1}{1.41}\right)^2 \quad (.75\ Zo + .75\ Zo + .5\ ZO)$$

The stub capacitance is also effectively decreased by the square of the turns ratio to lessen the loading problem. The 1:1.41 ratio of 1553B is a compromise between stub matching and decreased stub loading.

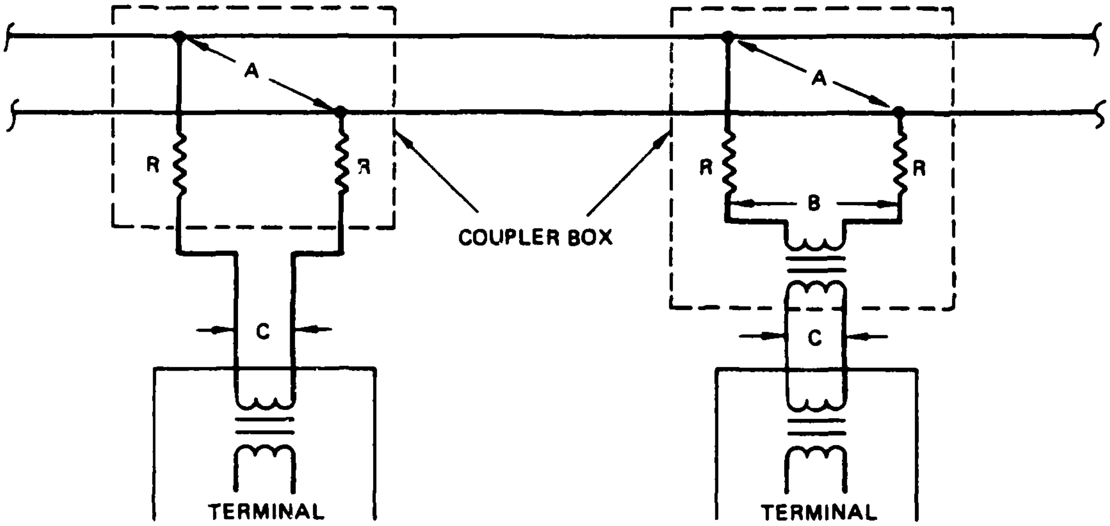
The connector type specified is important for severe environment military aircraft applications. MIL-STD-1553A specified the use of two-pin polarized connectors such as TEI BJ37 (reference to "TEL-14949-E137" is in error). The two-pin polarized connector employs an interface configuration with one male and one female contact. The female contact is embedded in one side of a step dielectric and the male contact is exposed. There are several inherent shortcomings to this design. MIL-STD-1553B does not specify connector types.

The coupling network provides bus connections for the transformer-coupled (external coupler) and direct-coupled cases defined in 1553B. Isolation resistors of 55 ohms value are included for the direct-coupled connection, and the proper transformer turns ratio is provided when the appropriate bus connection is selected. The turns ratio is different for the transformer-coupled and direct-coupled connections to compensate for the 1.41 to 1 reduction of signal level in the external coupler. This feature allows a threshold setting that is the same for both bus connections.

4.5.2 Terminal Characteristics

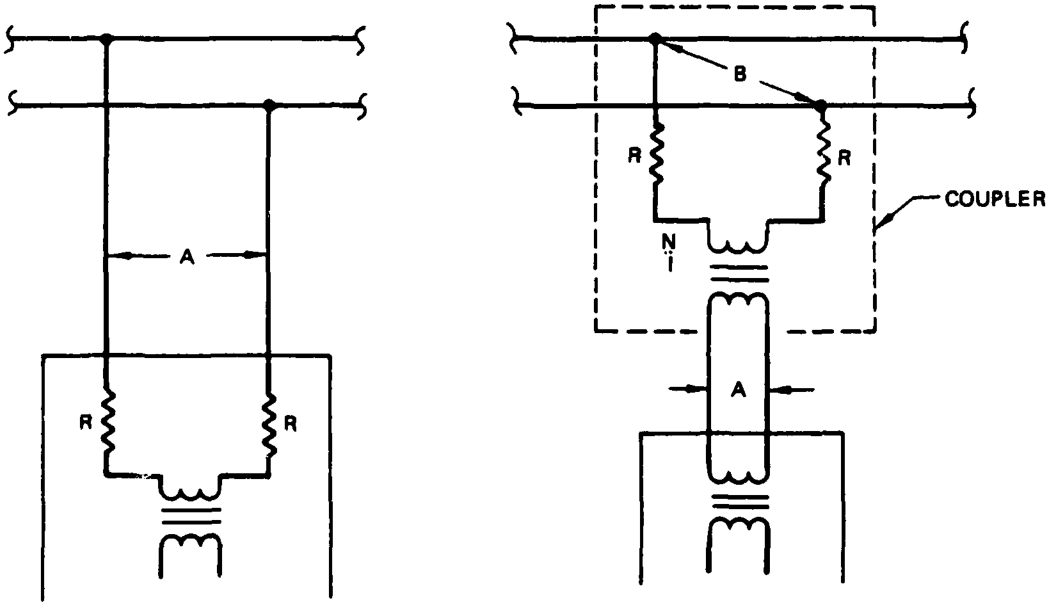
An additional concern is the specification for the bus and terminal interface. This area of 1553A was significantly reworked to provide a more complete definition of the terminal interface characteristics that are independent of network configuration. Figures 7-14 and 7-15 show the interface diagrams and the points where the signal measurement are defined in 1553A and 1553B. Table 7-5 is a summary listing of the terminal and data bus interface requirements specified in the two versions of the standard. The following discussion will relate some of the rationale for this approach to development of the updated requirements in 1553B.

4.5.2.1 Terminals with transformer coupled stubs



R Isolation resistor

Figure 7-14. MIL-STD-1553A Data Bus Interface



a. Direct Coupled (Short Stub) b. Transformer Coupled (Long Stub)

R isolation resistor
N transformer turns ratio

Figure 7-15. MIL-STD-1553B Data Bus Interface

Table 7-5. Summary of Terminal and Data Bus Interface Requirements

Parameter	MIL-STD-1553A	MIL-STD-1553B
Terminal output characteristics		
● Output voltage	±3.0V to ±10.0V, peak, l-l (6.0V to 20.0V, p-p, l-l) Point A, figure 4.2-4	18.0V to 27.0V, p-p, l-l (transformer coupled) Point A, figure 4.2-6B 6.0V to 9.0V, p-p, l-l (direct coupled) Point A, figure 4.2-6A
● Output waveform	Zero crossing deviation ± 25 ns; rise and fall time of this waveform shall be ≥100 ns	Zero crossing deviation ± 25 ns, maximum, measured with respect to previous crossing; rise and fall times 100 to 300 ns Overshoot and ringing ±900-mV peak, maximum, l-l Point A, figure 4.2-6B (transformer-coupled stub) Zero crossing deviation and rise and fall times same as above overshoot and ringing—±300-mV peak, maximum, l-l Point A, figure 4.2-6A (direct-coupled stub)

Table 7-5. *Summary of Terminal and Data Bus Interface Requirements (Continued)*

Parameter	MIL-STD-1553A	MIL-STD-1553B
● Output symmetry	Not specified	Voltage at 2.5 μs after midpoint of parity bit; ±250-mV peak, maximum, l-l Point A, figure 4.2-6B (transformer-coupled stub) Voltage at 2.5 μs after midpoint of parity bit; ±90-mV peak, maximum, l-l Point A, figure 4.2-6A (direct-coupled stub)
● Output Noise	10 mV p-p, l-l Point A, figure 4.2-4	14.0 mV, RMS, l-l Point A, figure 4.2-6B (transformer-coupled) 5.0 mV, RMS, l-l Point A, figure 4.2-6A (direct-coupled)
Terminal input characteristics		
● Input voltage	±0.5V to ±10.0V peak, l-l (1.0V to 20.0V, p-p, l-l) Point A, figure 4.2-4—terminal responds	0.86V to 14.0V, p-p, l-l, terminal response required; 0.0V to 0.2V, p-p, l-l, terminal no response (with transformer-coupled stubs) Point A, figure 4.2-5B 1.2V to 20.0V, p-p, l-l, terminal response required; 0.0V to 0.28V, p-p, l-l, terminal no response (with direct-coupled stubs) Point A, figure 4.2-5A

Table 7-5. Summary of Terminal and Data Bus Interface Requirements (Concluded)

Parameter	MIL-STD-1553A	MIL-STD-1553B
• Input impedance	2,000 ohms, minimum, from 100 kHz to 1.0 MHz Point C, figure 4.2-4	1,000 ohms, minimum, from 75 kHz to 1.0 MHz Point A, figure 4.2-5B (transformer-coupled stub) 2,000 ohms, minimum, from 75 kHz to 1.0 MHz Point A, figure 4.2-5A (direct-coupled)
• Noise rejection	BER–1 in 10^{12}; maximum, incomplete message rate 1 in 10^6; test condition–bus controller connected to RT over 100-ft data bus using 20-ft stubs	Maximum word error rate of 1 in 10^7, AWG noise, 1.0 kHz to 4.0 MHz, 140 mV, RMS level; signal level- 2.1V, p-p, l-l Point A, figure 4.2-5B (transformer -coupled stub) Maximum word error rate of 1 in 10^7, AWG noise, 1.0 kHz to 4.0 MHz, 200 mV, RMS level; signal level is 3.0V, p-p, l-l Point A, figure 4.2-5A (direct-coupled stub)
• Common mode rejection	±10.0V peak, line-to-ground, dc to 2 MHz, shall not degrade performance Point A, figure 4.2-4	±10.0V peak, line-to-ground, dc to 2.0 MHz, shall not degrade performance of the receiver Point A, figure 4.2-5B (transformer-coupled stub) Same specification for direct-coupled stub Point A, figure 4.2-5A

4.5.2.1.1 Terminal output characteristics. The following characteristics shall be measured with R_L, as shown on figure 12, equal to 70.0 Ohms ±2.0 percent.

4.5.2.1.1.1 Output levels. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 18.0 to 27.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

The upper end of the bus voltage range (20V p-p) allowed by 1553A was considered to be excessive and if implemented would result in excessive power dissipation and most of the systems and hardware designed to 1553A use signal levels at or near the lower end (6.0V p-p) of the specified range. It should be noted that the measurement point in 1553A is at the main bus, point A on figure 7-14. This does not provide a specified level at the terminal connection point (C) and is especially troublesome to the terminal hardware designer since the characteristics of the coupler transformer are not specified. The approach taken for 1553B is to specify the terminal output for the two conditions, transformer-coupled and direct-coupled. This may require that each terminal have two sets of input-output pins for each bus cable connection. Therefore, the 18V to 27V p-p transmitter output applied to the stub and coupler results in a nominal 6.0V to 9.0V p-p signal level at the stub and bus connection (point B). This range is equivalent to that specified for the direct-coupled case shown in figure 7-15. Test configurations are provided for both direct-coupled and transformer-coupled in figure 7-16.

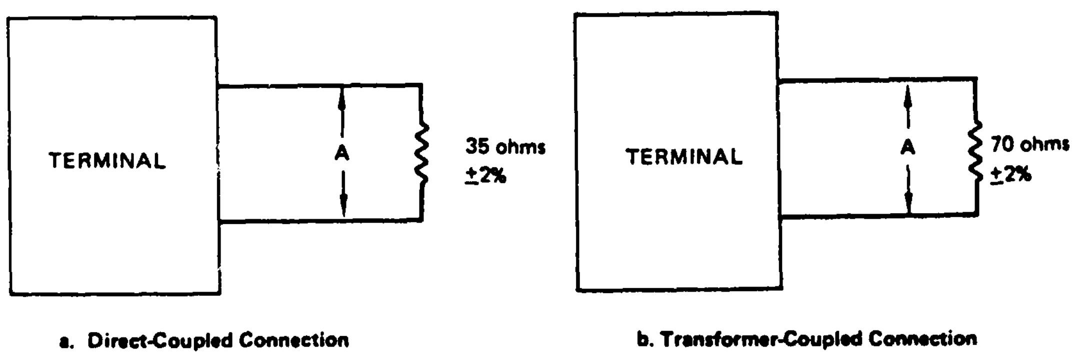


Figure 7-16. Direct-Coupled and Transformer-Coupled Terminal Output Test Configuration

4.5.2.1.1.2 Output waveform. The waveform, when measured at point A on figure 12 shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 us, 1.0 ± .025 us, 1.5 ± .025 us, and 2.0 ± .025 us). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 900.0 millivolts (mV) peak, line-to-line, as measured at point A, figure 12.

The transmitted waveform specified in 1553A is limited in the definition of signals that appear on the data bus. The zero crossing deviations allowed are not well defined for all possible patterns, and the rise and fall times specification is open ended. The waveform characteristics defined in 1553B provides control of the zero crossing deviations for all possible conditions and establishes a limit on distortion.

4.5.2.1.1.3 Output noise. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed a value of 14.0 mV, RMS, line-to-line, as measured at point A, figure 12.

MIL-STD-1553A specified value of 10 mV p-p noise (4.2.5.3.3) is considered unrealistically low for practical hardware design. Also, noise is normally specified as an rms value since peak noise is difficult to measure. The output rms noise for the transformer-coupled and direct-coupled cases are specified in 1553B (4.5.2.1.1.3 and 4.5.2.2.1.3) and are consistent with the required system performance and practical terminal hardware design. The requirement for low output noise of 14 mV rms and 5 mV rms when not transmitting also places significant constraints on the length and routing of input-output wiring because of the induced power supply and logic noise generated in the terminal.

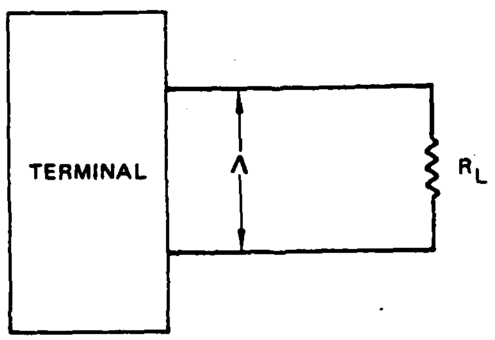


Figure 12 of 1553B. Terminal I/O Characteristics for Transformer-Coupled and Direct-Coupled Stubs

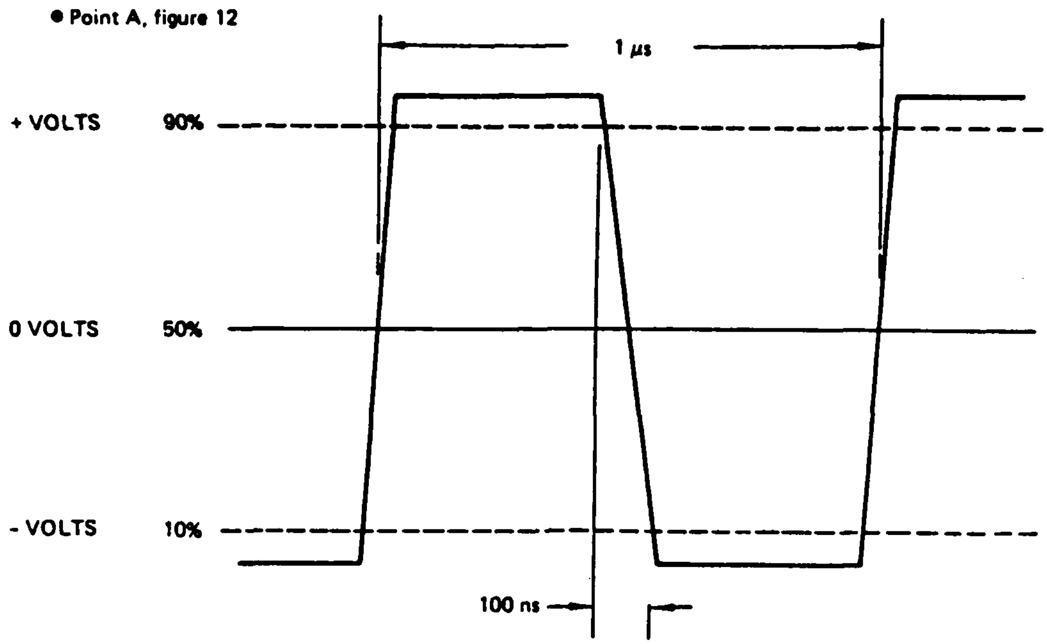


Figure 13 of 1553B. Output Waveform

4.5.2.1.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A of figure 12 shall be no greater than ±250.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are 8000_{16}, $7FFF_{16}$, 0000_{16}, $FFFF_{16}$, 5555_{16}, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.1.1.1 and 4.5.2.1.1.2.

Symmetry of the transmitted waveform in time and amplitude was not adequately specified in 1553A. An ideal waveform is perfectly balanced so that the signal energy on both sides of zero (off) level is identical. If the positive or negative energy is not equal, problems can develop in the coupling transformers and the transmission line can acquire a charge that appears as a tail with overshoot and ringing when transmission is terminated. These considerations require that the symmetry of the transmitted waveform be controlled within practical limits. This is accomplished in 1553B by specifying the signal level from a time beginning 2.5 us after the midbit zero crossing of the parity bit of the last word in a message transmitted by the terminal under test. The test messages contain the maximum number of words and defined bit patterns.

4.5.2.1.2 Terminal input characteristics. The following characteristics shall be measured independently.

4.5.2.1.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of ±150 ns, (i.e., 2.0 ± .15 us, 1.5 ± .15 us, 1.0 ± .15 us, .5 ± .15 us). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of .86 to 14.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .20 V. The voltages are measured at point A on figure 9.

4.5.2.1.2.2 Common mode rejections. Any signals from direct current (DC) to 2.0 MHz, with amplitudes equal to or less than ± 10.0 V peak, line-to-ground, measured at point A on figure 9, shall not degrade the performance of the receiver.

The input voltage specifications in 1553B have been revised to reflect the output voltage ranges for the transformer-coupled and direct-coupled connections to the terminal. The terminal-required response and no-response signal levels are specified so that the optimum threshold levels may be selected. It should be noted that the threshold setting has a significant effect on the noise rejection and error rate performance of the receiver. The maximum value for no-response signal level is 200 mV p-p, and 280 mV p-p allows optimum threshold settings of ±125 and ±175 mV, respectively, for minimum bit error rate (BER) performance when subjected to the specified noise rejection test conditions.

4.5.2.1.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 1000.0 Ohms within the frequency range of 75.0 kHz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 9.

As indicated in the data bus network requirement, input impedance is required to be maintained at a reasonable level to reduce the signal distortion effects when terminals are connected to the bus. Terminal input impedance is determined primarily by the following:

a. Transformer impedance maintains inductance required to support low-frequency component of signal while controlling interwinding capacitance for high frequencies.

b. Terminal wiring capacitance controls stray capacitance of wiring from terminal connector to receiver.

c. Secondary impedance transformation, for the transformer-coupled case, a transformer with a turns ratio of 1:1.41 is implied. The impedance at the secondary is reflected to the terminal input reduced by a factor of 2.

The transformer is a very important element in determining the transceiver characteristics such as input impedance, signal waveform integrity, and

common mode rejection required by 1553B. The considerations for transformer and associated input-output circuit design are to--

a. provide the specified input impedance at high frequencies (terminal input impedance 1,000 ohms and 2,000 ohms at 1 MHz)

b. Design for low interwinding capacitance to achieve the common mode rejection (45 db CMR at ± 10V peak, dc to 2.0 MHz)

These considerations are directly applicable to the design of the transceiver transformer. In addition to the transformer characteristics, other considerations for maintaining the terminal minimum input impedance specified in 1553B are as follows:

a. Minimize stray capacitance of wiring from the external connector and on the circuit card to the buffer amplifier (every 100 Pf results in approximately 1,600 ohms of shunt impedance).

b. Maintain high impedance at the receiver limiter and filter circuit inputs and transmitter driver outputs in the "off" state. These impedances must be maintained with the terminal (transceiver) power off.

The factor of 2 difference in the impedance specified for the transformer-coupled and direct-coupled cases is based primarily on the effect of c. The frequency range was changed to reduce the lower frequency limit from 100 kHz (1553A) to 75 kHz (1553B). This provides additional assurance that adequate transformer volt-time product (inductance) is available to support the lower frequencies of the signal without approaching saturation.

> 4.5.2.1.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in 10^7, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 140 mV. A word error shall include any fault which shall be measured with a 2.1 V peak-to-peak, line-to-line, input to the terminal as measured at point A on figure 9. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

The noise rejection specification and test conditions defined in 1553A requires extensive system-type evaluation testing of the terminal employing a bus controller and data bus radiated with certain of the EMI fields specified in MIL-STD-461 and 462. Extensive test time is required to verify a BER of 10^{-12} and the test must be performed in a screen room.

Table II. Criteria for Acceptance or Rejection of a Terminal for the Noise Rejection Test

Total words received by terminal (in multiples of 10^7)

No. of errors	Reject (equal or less)	Accept (equal or more)
0	NA	4.40
1	NA	5.21
2	NA	6.02
3	NA	6.83
4	NA	7.64
5	NA	8.45
6	0.45	9.27
7	1.26	10.08
8	2.07	10.89
9	2.88	11.70
10	3.69	12.51
11	4.50	13.32
12	5.31	14.13
13	6.12	14.94
14	6.93	15.75
15	7.74	16.56
16	8.55	17.37
17	9.37	18.19
18	10.18	19.00
19	10.99	19.81
20	11.80	20.62
21	12.61	21.43
22	13.42	22.24
23	14.23	23.05
24	15.04	23.86
25	15.85	24.67
26	16.66	25.48
27	17.47	26.29
28	18.29	27.11
29	19.10	27.92
30	19.90	28.73
31	20.72	29.54
32	21.53	30.35
33	22.34	31.16
34	23.15	31.97
35	23.96	32.78
36	24.77	33.00
37	25.58	33.00
38	26.39	33.00
39	27.21	33.00
40	28.02	33.00
41	33.00	NA

Note: NA—not applicable.

The test conditions of signal and noise specified in 1553B were selected to produce a corresponding value of word error ratio (WER) which is sufficiently high (10^{-7}) to permit performance verification of a terminal receiver within a reasonable test period. The noise rejection is a figure-of-merit test and can be performed in a normal laboratory environment using actual transmitters (Manchester waveform output) with a typical test setup as shown in figure 7-17. The verification of detector performance should consider the measurement of both detected and undetected errors. To measure undetected errors that do not correlate with the transmitted signal and are not detected by the terminal under test, it is necessary to compare the transmitted and received data. Therefore, a reference of transmitted data is provided to the comparator for comparison with the detected data from the terminal under test.

Externally generated noise on board an aircraft can take many forms with a wide variety of power and frequencies. It is recognized that impulse noise having either random or periodic impulse duration, frequency of occurrences, and burst interval are more typical of noise sources that have major impact on aircraft digital data systems. Relay switching is generally regarded as the most severe source of impulse noise on a typical aircraft. This type of noise defies accepted forms of analysis, such as that performed utilizing additive white gaussian (AWG) noise model. Because of the difficulty of error performance analysis using the impulsive noise model, a worst-case gaussian model has been formulated. This model offers an analysis and test tool for evaluation of terminal receiver performance considering the effects of impulsive noise. This approach is reflected in the noise rejection test conditions and word error rate versus signal-to-noise ratio (SNR) performance requirements of 1553B, paragraphs 4.5.2.1.2.4 and 4.5.2.2.2.4.

4.5.2.2 <u>Terminals with direct coupled stubs</u>

4.5.2.2.1 <u>Terminal output characteristics</u>. The following characteristics shall be measured with R_L, as shown on figure 12, equal to 35.0 Ohms ± 2.0 percent.

4.5.2.2.1.1 <u>Output levels</u>. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 6.0 to 9.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

4.5.2.2.1.2 <u>Output waveform</u>. The waveform, when measured at point A on figure 12, shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 us, 1.0 ± .025 us, 1.5 ± .025 us and 2.0 ± .025 us). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 300.0 mV peak, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.3 <u>Output noise</u>. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed

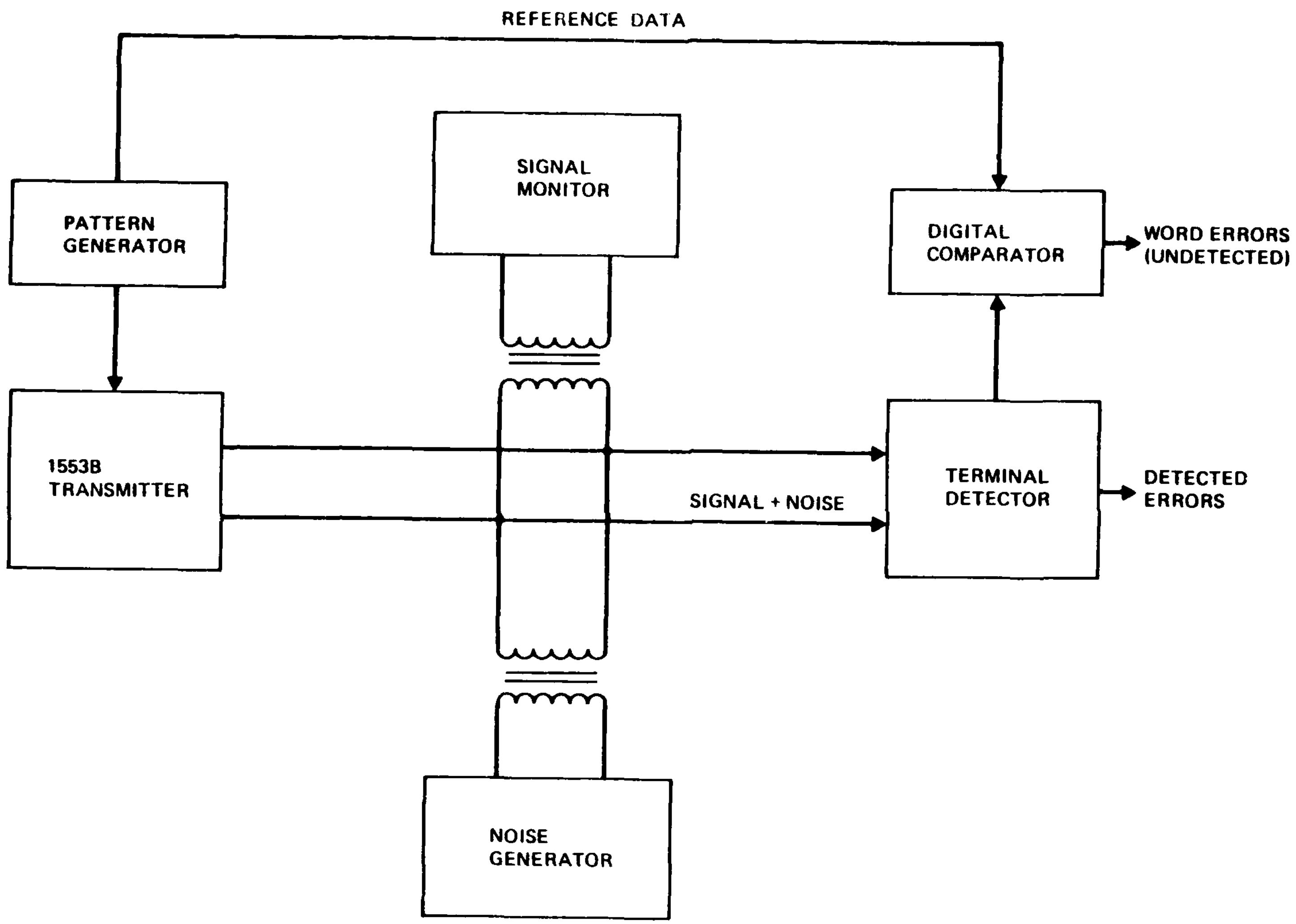


Figure 7-17. Typical Noise Rejection Test Setup

a value of 5.0 mV, RMS, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A on figure 12, shall be no greater than ± 90.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are 8000_{16}, $7FFF_{16}$, 0000_{16}, $FFFF_{16}$, 5555_{16}, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.2.1.1 and 4.5.2.2.1.2.

4.5.2.2.2 Terminal input characteristics. The following characteristics shall be measured independently.

4.5.2.2.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of plus or minus 150 ns, (i.e., 2.0 ± .15 us 1.5 ± .15 us, 1.0 ± .15 us .5 ± .15 us). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 1.2 to 2.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .28 V. The voltages are measured at point A on figure 10.

4.5.2.2.2.2 Common mode rejections. Any signals from DC to 2.0 MHz, with amplitudes equal to or less than ± 10.0 volts peak, line-to-ground, measured at point A on figure 10, shall not degrade the performance of the receiver.

4.5.2.2.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 2000.0 ohms within the frequency range of 75.0 kHz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 10.

4.5.2.2.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in 10^7, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 200 mV. A word error shall include any fault which causes the message error bit to be set in the terminal's status word, or one which causes a terminal to not respond to a valid command. The word error rate shall be measured with a 3.0 V peak-to-peak, line-to-line, input to the terminal as measured at point A on

figure 10. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

A discussion of "terminals with direct coupled stubs" are covered in the preceding section, which discusses "terminals with transformer coupled stubs" (4.5.2.1).

4.6 Redundant data bus requirements. If redundant data buses are used, the requirements as specified in the following shall apply to those data buses.

This section of the standard was added to 1553B. The purpose was to clarify requirements relating to the electrical characteristics and operation of redundant data buses.

4.6.1 Electrical isolation. All terminals shall have a minimum of 45 dB isolation between data buses. Isolation here means the ratio in dB between the output voltage on the active data bus and the output voltage on the inactive data bus. This shall be measured using the test configuration specified in 4.5.2.1.1 or 4.5.2.2.1 for each data bus. Each data bus shall be alternately activated with all measurements being taken at point A on figure 12 for each data bus.

Although the data bus is commonly used in a dual-redundant manner, 1553A did not specify electrical isolation characteristics between redundant buses. This paragraph was added in 1553B to provide a ratio of the maximum transmitted signal on one bus to the minimum received threshold on the redundant bus. Test conditions that are specified in the referenced paragraphs correspond to terminals with transformer-coupled stubs and with direct-coupled stubs.

4.6.2 Single event failures. All data buses shall be routed to minimize the possibility that a single event failure to a data bus shall cause the loss of more than that particular data bus.

This is an obvious requirement unique to military aircraft. This paragraph was added in 1553B.

4.6.3 Dual standby redundant data bus. If a dual redundant data bus is used, then it shall be a dual standby redundant data bus as specified in the following paragraphs.

4.6.3.1 Data bus activity. Only one data bus can be active at any given time except as specified in 4.6.3.2.

4.6.3.2 Reset data bus transmitter. If while operating on a command, a terminal receives another valid command, from either data bus, it shall reset and respond to the new command on the data bus on which the new command is received. The terminal shall respond to the new command as specified in 4.3.3.8.

These new paragraphs in 1553B reflect the common practice in current aircraft of using the dual bus as one active, one standby, and it was the intent to restrict the operation of a dual data bus connected to terminals to a "one-at-a-time" operation. However, provision had to be made for a bus controller to override one bus to respond on the redundant bus. The requirement for this is in paragraph 4.6.3.2 above, and the reference to paragraph 4.3.3.8 is the response time requirement of a remote terminal to a valid command word.

CHAPTER 8

DEFINITIONS

8.0 TERMS AND DEFINITIONS

1553. Military standard that establishes requirements for digital, command/response time-division multiplexing techniques on aircraft. In this handbook, 1553 is used as a generic name for MIL-STD-1553(USAF), MIL-STD-1553A, and MIL-STD-1553B. Where a specific revision is referenced, the revision suffix is added, e.g., 1553B.

Aperiodic. Events occurring at indefinite or unscheduled time periods. This term is used to describe the timing of messages that are not assigned a regular transmission update rate. "Asynchronous" is another word used to express the same condition..

Applications Software. Those programs whose purpose is dedicated to specific functions (e.g., navigation, fire control, I/O conversion). This distinguishes applications software from executive software.

Architecture. The functional structure of a multiplex system as defined by the physical interconnection (topology) and the multiplex system control. See configuration.

Asynchronous. Without timing regularity, asynchronous operation per 1553B is: "For the purpose of the standard, asynchronous operation is the use of an independent clock source in each terminal for message transmission. Decoding is achieved in receiving terminals using clock information derived from the message." In this handbook, this term has not been used to describe the timing of messages except for its use by DAIS in section 6.5. See aperiodic.

Asynchronous Operation. For the purpose of the standard, asynchronous operation is the use of an independent clock source in each terminal for message transmission. Decoding is achieved in receiving terminals using clock information derived from the message.

Avionics. All electronic and electromechanical systems and subsystems (hardware and software) installed in an aircraft or attached to it.

Bit. Contraction of binary digit--may be either zero or one. In information theory, a binary digit is equal to one binary decision or the designation of one of two possible values or states of anything used to store or convey information.

Bit Rate. The number of bits transmitted per second.

Built-in Test (BIT). The capability of an LRU to perform some form of self-test.

Bus Interface Unit (BIU) Function. This term is generally interchangeably with "terminal," as defined in 1553B: "The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate line replaceable units (LRUs) or be contained within the elements of the subsystem."

Bus Interface Unit (BIU) Hardware. This term describes a particular set of hardware that performs the interface between the data bus and the internal

portion of an embedded or standalone remote terminal. As a minimum, it refers to the digital decode and encode logic that expands to the complete analog-to-digital interface between the data bus and the internal remote terminal electronics or the subsystem for embedded terminals.

Broadcast. Operation of a data bus system such that information transmitted by the bus controller or a remote terminal is addressed to more than one of the remote terminals connected to the data bus.

Bus. In this handbook (unless noted in the text) bus refers to 1553 data bus. See data bus.

Bus Controller. The terminal assigned the task of initiating information transfers on the data bus. Note that an "information transfer" is a format that includes transmission and response by the appropriate terminals.

Bus Monitor. The terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time.

Command/Response. Operation of a data bus system such that remote terminals receive and transmit data only when commanded to do so by the bus controller.

Communications Protocol (protocol). The conventions imposed on serial data to ensure that the receiver correctly interprets the transmitted data; also, the procedures used for initiating messages and responding to them.

Compool (global/local) (communication pool). This is a JOVIAL language term used to declare or define data by name that will subsequently be set or used by program procedures. The JOVIAL compiler establishes locations in memory for compool data. As such, it is the means of data communications, and the scope of the declaration can be limited, hence the restriction "local."

Configuration. The specific functional structure of a given integrated system consisting of physical interconnection (topology) and system control. See architecture.

Data. This term is used in this handbook to denote the content (16 bits) of information transferred on the 1553 data bus in one data word.

Data Bus. Whenever a data bus or bus is referenced in this handbook, it implies all the hardware, including twisted-shielded pair cables, isolation resistors, transformers, etc., required to provide a single data path between the bus controller and all the associated remote terminals.

Direct Coupled. A method of connecting terminals to the 1553 data bus using only a wire splice.

Dynamic Bus Control. The operation of a data bus system in which designated terminals are offered control of the data bus.

Embedded Interface. 1553 interface circuitry housed within a subsystem.

Error Recovery. General terms used to describe the detection of transients or hard failures and the step-by-step sequence to branch to alternate functions, procedures, or equipment use.

Event. A single occurrence at a precise time.

Function. The special work done by a subsystem or a software task.

Half Duplex. Operation of a data transfer system in either direction over a single line but not in both directions on that line simultaneously.

Hierarchical Control. A form of distributing all system control in a system, where one level of control is subordinate to a higher level of control.

Hierarchical Network. A description of a physical topology that has both global and local levels of data buses.

Integration. In this handbook, integration refers to the cooperative need for shared information and the means for achieving that cooperation.

Input-Output (I/O). This term is used to describe both the function of hardware and software to receive and transmit data and the physical hardware section that is the interface between a 1553 interface and subsystems of a remote terminal or bus controller.

Major Cycle. A period of scheduled time during which all periodic transmissions and computations occur at least once. Major cycles are divided into subcycles called minor cycles.

Message. In 1553 terms, a message is a part of an information transfer format, such as 1 to 32 data words. A message may also refer to the entire transmission by both bus controller and responding remote terminal. which includes not only the data words but the overhead. This second usage is more correctly called an information transfer format.

Definition from 1553: "A single message is the transmission of a command word, status word, and data words if they are specified. For the case of a remote terminal to remote terminal (RT to RT) transmission, the message shall include the two command words, the two status words and data words."

Minor Cycle. A period of scheduled time during which the most frequently occurring periodic transmission or computation will occur, or a period scheduled for a frequently occurring transmission or computation. Multiple minor cycles may be required to achieve a major cycle. See major cycle.

Mode Code. A means by which the bus controller can communicate with the multiplex-bus-related hardware to assist management of the information flow.

Modem. In this handbook, this term is used to mean the analog transceiver circuitry used to convert to digital form. It is also used to denote a bus interface unit function.

Modulation. The signaling method used to convey data on the data bus.

Multiplexing. The transmission of information from several signal sources through one communication system.

Partitioning. The method used to divide a complex system or function into manageable size before allocating these smaller pieces to devices to perform the required job.

Periodic. Event(s) recurring at specific time intervals. See aperiodic.

Protocol. The conventions imposed on serial data to ensure that the receiver correctly interprets the transmitted data; also, the procedures used for initiating messages and responding to them. See communications protocol.

Pulse Code Modulation (PCM). The form of modulation in which the modulation signal is sampled, quantized, and coded so that each element of information consists of different types or numbers of pulses and spaces.

Redundancy. The method of replicating a function for the purpose of increasing the availability of the function.

Redundant Data Bus. The use of more than one data bus to provide more than one data path between the subsystems (i.e., dual-redundant data bus and triredundant data bus).

Retrofitting. The process of updating a system with new or modernized equipment.

Remote Terminal (RT). All terminals not operating as the bus controller or as a bus monitor.

Sensor. The hardware and software required to perform a specific avionic function.

Specification. A document prepared specifically to support procurement that clearly and accurately describes the essential technical requirements for purchased material. Also included are procedures necessary to determine that the requirements have been met for the purchased material covered by the document.

Standard. A military standard is a document that establishes engineering and technical requirements for processes, procedures, practices, and methods that have been adopted as standard.

Stub. The electrical interface between the data bus and the terminal.

Subsystem. The hardware and software required to perform a specific avionic function. See sensor.

Definition from 1553: "The device or functional unit receiving data transfer service from the data bus."

Synchronous. Events occurring at specific time intervals. See aperiodic.

System. The collection of subsystem equipment required to perform a specific mission or job.

System Control. The local control of each of the data bus elements as well as the coordination of this control to perform an orderly functioning system.

Terminal. The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate LRUs or be contained within the elements of the subsystem.

Time Division Multiplexing (TDM). The transmission of information from several signal sources through one communication system with different signal samples staggered in time to form a composite pulse train.

Topology. The interconnectivity of the data bus(es) and their associated elements (terminals and controllers) to accomplish the desired data path required by the integration.

Transformer Coupled. A method of stubbing to the 1553 data bus that uses a transformer and isolation resistors.

Word. A 1553 word is a sequence of 20 bit times consisting of a 3 bit-time sync, 16 bits of data, and 1 parity bit. This is the word as it is transmitted on the bus; 1553 terminals add the sync and parity before transmission and remove them during reception. Therefore, the nominal word size is 16 bits, most significant bit first. There are three types of words: command, status, and data.

Definition from 1553B: In this document, a word is a sequence of 16 bits plus sync and parity. There are three types of words: command, status, and data.

CHAPTER 9

BIBLIOGRAPHY

9.0 BIBLIOGRAPHY

Allen, R., et al., "System Avionic Architectures for RPVs," Document AFAL-TR-76-245, Texas Instruments, Inc., Dallas, Texas, April 1977.

Anderson, W., "F-16 Avionics Multiplex Data Bus," AFSC Multiplex Data Bus Conference Proceedings, General Dynamics Convair, Fort Worth, Texas, November 1976.

Anderson, W., "F-16 Avionic Multiplex Data Bus System," presentation at SAE/A-2K Subcommittee Meeting No. 17, General Dynamics, Fort Worth, Texas, April 7-8, 1976.

Barber, G. W., and Rich, B. A., "Compatibility Testing and Integration of DAIS Multiplex Bus Equipments," NAECON '77 Record, TRW, Redondo Beach, California, May 1977.

Bona, B. E., "Microprogrammable Data Terminal," AFSC Multiplex Data Bus Conference Proceedings, Autonetics Group, Rockwell International, Anaheim, California, November 1976.

Bona, B. E., "Some Applications of the Rockwell Data Terminal Chip," Second AFSC Multiplex Data Bus Conference Proceedings, Autonetics Strategic Systems, Rockwell International, Anaheim, California, October 1978.

Boose, Emery F., "Lessons Learned Through a MIL-STD-1553 Time Division Multiplex Bus," NAECON '75 Record, MITRE Corp., Bedford, Massachusetts, June 1975.

Boose, Costa, and Mahler, "Shielded Twisted Pair as a Transmission Medium in a Multisubscriber Time Division Multiplex System," NAECON '75 Record, MITRE Corp., Bedford, Massachusetts, June 1975.

Booth, Frank, and Espen, David, "Multiplex System for the Hughes Advanced Attack Helicopter--YAH-64," Second AFSC Multiplex Data Bus Conference Proceedings, Hughes Helicopters, Inc., Culver City, California, and Sperry Flight Systems, Phoenix, Arizona, October 1978.

Brickner, David R., "Military Multiplex Standard Defines Versatile Serial Bus," Computer Design, Sperry Flight Systems, Phoenix, Arizona, December 1979.

Ciasulli, L., and Henderson, P., "ULAIDS--1553 Architecture with Dynamic Bus Allocation," Second AFSC Multiplex Data Bus Conference Proceedings, Conrac Corporation, West Caldwell, New Jersey, October 1978.

Cotter, G. E., and Fitzgerald, F., "Avionics Multiplexing with Smart Terminals," AFSC Multiplex Data Bus Conference Proceedings, Grumman Aerospace, November 1976.

Crossgrove, W. A., and Smith, Dr. L. A., "IDAMST: Integrated Digital Avionics for the Advanced Medium STOL Transport," AFSC Multiplex Data Bus Conference Proceedings, Boeing Aerospace Company, Seattle, Washington, November 1976.

Desipio, Richard G., LAMPS MKIII Avionic System Description, U.S. Navy, Naval Air Development Center (NADC), November 1978.

Desipio, Richard G., F-18 Mission Computer Multiplex System, Naval Air Development Center, November 1978.

Ellis, D. H., "Multiplex Interface Technical Review," AFSC Multiplex Data Bus Conference, SCI Systems, Inc., Huntsville, Alabama, November 1976.

Engelland, James D., et al., Operational Software Concept (Phase Two), Document AFAL-TR-75-230, General Dynamics, Fort Worth, Texas, January 1976.

Gangl, E. C., "Time Division Multiplexed Data Bus Integration Techniques," AFCS Multiplex Data Bus Conference, ASD/ENAI, WPAFB, Ohio, November 1976.

Gross, J. P., Jr., "Data Bus Techniques for Digital Avionics," NAECON '73 Record, SCI Systems, Inc., Huntsville, Alabama, May 1973.

Gutierrez, A., and Pritchard, Maj. G., "The Design and Development of an LSI Bus Interface Unit," Second AFSC Multiplex Data Bus Conference Proceedings, AFAL/WPAFB, Ohio, October 1978.

Hermann, W. S., "Multiplex Transmission Methods," No. 115, DFCS Newsletters, McDonnell Aircraft Co., St. Louis, Missouri, November 1974.

Hermann, W. S., "Multiplex Data Bus Operation," No. 265, DFCS Newsletters, McDonnell Aircraft Co., St. Louis, Missouri, December 1974.

Hermann, W. S., "Multiplex Terminal Operation," No. 167, DFCS Newsletters, McDonnell Aircraft Co., St. Louis, Missouri, February 1975.

Hermann, W. S., and Smith, K. C., "A Redundant Multiplex Data Bus for F-18," AFSC Multiplex Data Bus Conference Proceedings, McDonnell Aircraft Co., St. Louis, Missouri, November 1976.

Hoffman, Robert, "Dual Channel 1553A Interface," Second AFSC Multiplex Data Bus Conference Proceedings, Applied Technology, Division of Itek Corp., Sunnyvale, California, October 1978.

Hollowich, M. E., and McClimens, M. G., The Software Design and Verification System (SDVS), Document AFAL-TR-76-200, TRW, Redondo Beach, California, May 1977.

Hunter, W. R., "A Hybridized 1553A Interface," Second AFSC Multiplex Data Bus Conference Proceedings, Teledyne Systems Co., October 1978.

Husbands, Charles R., "Microprocessors in Airborne Time Division Multiplex Terminals," NAECON '75 Record, MITRE Corp., Bedford, Massachusetts, June 1975.

Lambert, J. W., and Green, J. P., A Comparative Study of Data Bus Systems, Document IR-270, Intermetrics, Inc., Cambridge, Massachusetts, March 22, 1978.

McCoy, Bruce J., "New Flight Software Development Techniques--The DAIS Example," NAECON '75 Record, C. S. Draper Laboratory, Cambridge, Massachusetts, June 1975.

McCuen, J. W., "Use of Standard Modules for Airborne Multiplexing Applications," Second AFSC Multiplex Data Bus Conference Proceedings, Hughes Aircraft, Microelectronics Division, Newport Beach, California, November 1976.

McCuen, J. W., and Smith, G. E., "Use of Standard Modules for Airborne Multiplexing Applications, Chapter 2," Second AFSC Multiplex Data Bus Conference Proceedings, Hughes Aircraft, Microelectronics Division, Newport Beach, California, October 1978.

McLaren, Barry D., "B-1 Avionics Data Multiplexing System (AMUX)," NAECON '75 Record, Boeing Aerospace Company, June 1975.

Ohlhaber, J. I. and Orlando, F. J., "Lessons Learned in Development of the Data Bus System," AFSC Multiplex Data Bus Conference Proceedings, Harris Corporation, EDS, Melbourne, Florida, November 1976.

Pensworth, Capt. Frederick L., "Digital Avionic Information System (DAIS) Multiplex System," AFSC Multiplex Data Bus Conference Proceedings, AFAL/WPAFB,
Ohio, November 1976.

Peter, F. E., LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Block Diagram," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Word Formats," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Memory Map," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Drawings (Logic) #891AS555, 8 Sheets," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Multiplex Terminal Unit, #891AS578, 1 Sheet," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Flow Diagrams (16)," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Peter, F. E., "Computer Printout (Program Listing)," LSI-11 BCIU Design Information, Naval Avionics Center, Indianapolis, Indiana, November 1978.

Rogers, R. T., "Application of LSI and Hybrid Technology to Avionic Multiplexing," AFSC Multiplex Data Bus Conference Proceedings, Singer-Kearfott Division, Singer Corporation, November 1976.

Salter, Robert M., "1553B: How Do You Know You'll Pass the Noise Test?," Second AFSC Multiplex Data Bus Conference Proceedings, Sperry Univac, St. Paul, Minnesota, October 1978.

Schocket, F., and Hungerford, R., "The Application of MIL-STD-1553A to the LAMPS MK-III Airborne System," Second AFSC Multiplex Data Bus Conference Proceedings, Naval Air Development Center, Warminster, Pennsylvania, October 1978.

Shatas, R., and Blevins, J., "A Flexible Multiplex Terminal Interface Unit," Second AFSC Multiplex Data Bus Conference Proceedings, SCI Systems, Inc., Huntsville, Alabama, October 1978.

Simpson, Robert C., Design of the Bus Interface Unit for the Distributed Processor/Memory System, Report GE/EE/76D-40, Air Force Institute of Technology, WPAFB, Ohio, December 1976.

Sternberg, W. J., "Stores Management and Data Bus Systems," NAECON '77 Record, Delco Electronics Division, GMD Corp., Santa Barbara, California, May 1977.

Sundstrom, D. E, Anderson, W., and Alford, S. A., "F-16 Multiplex: A Systems Perspective," Second AFSC Multiplex Data Bus Conference Proceedings, General Dynamics, Fort Worth, Texas, October 1978.

Thing, R. L., Miller, L. G., and Geyer, M. A., "Operational Experience with a BCIU Integrated into a Processor," AFSC Multiplex Data Bus Conference Proceedings, Westinghouse Electric, Systems Development Division, Baltimore, Maryland, November 1976.

Trainor, W. Lynn, "Software for Avionics: An Overview," NAECON '75 Record, AFAL/WPAFB, Ohio, June 1975.

Turner, C. Ray, "Development of the MIL-STD-1553 Multiplex Applications Handbook," Second AFSC Multiplex Data Bus Conference Proceedings, Boeing Aerospace Company, Seattle, Washington, October 1978.

Turner, C. Ray, and McCall, Mack B., "Analysis and Evaluation of Current MIL-STD-1553 Digital Avionics Architectures as the Basis for Advanced Architectures Using MIL-STD-1553B," NAECON '79 Record, Boeing Aerospace Company, Seattle, Washington, May 1979.

Yelverton, D. G., "The Bus Interface Unit, an Adaptable LSI Chip-Set," Second AFSC Multiplex Data Bus Conference Proceedings, Harris Corporation, Melbourne, Florida, October 1978.

Zawodny, Thomas L., "Bus Controller Software Implementation," Second AFSC Multiplex Data Bus Conference Proceedings, ASD/ENAIA (SEAFAC), WPAFB, Ohio, October 1978.

LAMPS MK-III Avionic System Data Bus Interface Requirements, Detail Specification for Document 6226114C, IBM Federal Systems Division, Owego, New York.

Interface Characteristic Specification, LAMPS MK-III, Avionics Data Bus Controller and Avionics Document 6226054C, IBM Federal Systems Division, Owego, New York.

Military Standard Aircraft Internal Time Division Command/Response Multiplex Data Bus, MIL-STD-1553A, April 30, 1975.

Military Standard Aircraft Internal Time Division Command/Response Multiplex Data Bus, MIL-STD-1553B, September 21, 1978.

Excerpt from B-1 Avionics Interface Specification - AMUX Characteristics, Document NA-75-100A, Part II, Vol. I, pages 68-77, Boeing Aerospace Company, Seattle, Washington, December 5, 1975.

Development Specification for Input/Output Modem II, Document D229-10395-1, Model B-1, Boeing Aerospace Company, Seattle, Washington, December 9, 1974 (preliminary).

B-52 OAS Multiplex Bus Protocol, Document D675-10110-1, The Boeing Company, Seattle, Washington, April 10, 1978.

A Study of Multiplex Data Bus Techniques for the Space Shuttle, Final Summary Report, Report 2635-M15, prepared for NASA/MSFC by SCI Systems, Inc., Huntsville, Alabama, November 1972.

Interface Control Document for the F-16 Avionic System, Document 16PP188(B), General Dynamics, Fort Worth, Texas, February 18, 1976.

The Impact of Wideband Multiplex Concepts on Microprocessor-Based Avionic System Architectures, Final Report, Document AFAL-TR-78-4, AFAL/WPAFB, Ohio, February 1978.

Standard Multiplex Interface Definition for Advanced Aircraft Avionics, AFAL-TR-74-B2, AFAL/WPAFB, SCI Systems, Inc., Huntsville, Alabama, July 1974.

Multiplexing in Aircraft Subsystems, AFAPL-TR-73-41, AFAPL/WPAFB, SCI Systems, Inc., Huntsville, Alabama, July 1973.

The Application of Information Transfer Techniques for Solving the Internal Communication Requirements of a Night AX Aircraft, AFAL-TR-72-289, AFAL/WPAFB, SCI Systems, Inc., Huntsville, Alabama, March 1973.

Study and Development of a Standard Digital Interface, AFAL-TR-72-223, AFAL/WPAFB, Hughes Aircraft, Culver City, California, July 1972.

Control Document for Multiplex System Interface, CD14, 11D41, Hughes Helicopters, Inc., August 3, 1978.

Procurement Specification for the YAH Multiplex System, PS-14-11030B, Hughes Helicopters, Inc., September 18, 1978.

The Application of Multiplexing to the A-7D Aircraft, AFAL-TR-73-302, AFAL/WPAFB, Radiation Division of Harris-Intertype Corporation, September 1973.

The Application of Information Transfer Techniques for Solving the Internal Communication Requirements of an Advanced Manned Bomber, AFAL-TR-72-209, AFAL/WPAFB, Radiation Division of Harris-Intertype Corporation, September 1972.

Technical Description of the Digital Avionics Information System (DAIS), Document PA 100101, AFAL/WPAFB, Ohio, February 10, 1978.

DAIS System Specification, Document SA 100100, AFAL/WPAFB, Ohio.

System Control Procedures, Document MA 201200, AFAL/WPAFB, Ohio, March 7, 1978.

Data Bus Control Interface Unit, Document SA 301400, AFAL/WPAFB, Ohio.

Data Bus Control Interface Unit, Document SA 301300, AFAL/WPAFB, Ohio.

F-18 Avionics Multiplex Design Specification, Report MDC A3818B, McDonnell Aircraft Company, St. Louis, Missouri, October 1, 1976.

F-15 Avionics Multiplex Design Specification, Report MDC A4905, McDonnell Aircraft Company, St. Louis, Missouri, August 15, 1977.

Management Guide to Avionics Software Acquisition, Volume I: An Overview of Software Development and Management, Document ASD-TR-76-11-Vol-1, Logicon Inc., Dayton, Ohio, June 1976.

Management Guide to Avionics Software Acquisition, Volume II: Software Acquisition Process, Document ASD-TR-76-11-Vol-2, Logicon Inc., Dayton, Ohio, June 1976.

Management Guide to Avionics Software Acquisition, Volume III: Summary of Software Related Standards and Regulations, Document ASD-TR-76-11-Vol-3, Logicon Inc., Dayton, Ohio, June 1976.

Management Guide to Avionics Software Acquisition, Volume IV: Technical Aspects Relative to Software Acquisition, Document ASD-TR-76-11-Vol-4, Logicon Inc., Dayton, Ohio, June 1976.

Advanced Avionics Systems for Multimission Applications (AASMMA), Volumes 1 and 2, Interim Technical Report 1, AFAL/WPAFB, The Boeing Company, Military Airplane Development, Seattle, Washington, November 1978.

Multiplex Techniques Safety Study, AFWL-TR-73-38, Hughes Aircraft, June 1973.

Nuclear Safety Design Criteria for Weapons Systems Employing Digital Avionic Information Systems (DAIS), AFWL-TR-75-86, Hughes Aircraft, October 1975.

B-1 Unique Signal Train Generation Study, AFWL-TR-74-297, Boeing Aerospace Company, November 1974.

Nuclear Weapon Multiplex Control Interface, AFWL-TR-79-166, Hughes Aircraft, April 1980.

CHAPTER 10

MIL-STD-1553B

MIL-STD-1553B
21 September 1978
SUPERSEDING
MIL-STD-1553A
30 April 1975

MILITARY STANDARD

AIRCRAFT INTERNAL TIME DIVISION

COMMAND/RESPONSE MULTIPLEX DATA BUS

FSC MISC

MIL-STD-1553B
21 Septemb•r 978

DEPARTMENT OF DEFENSE
Washington D.C. 20360

Aircraft Internal Time Division Command/Response Multiplex Data Bus

MIL-STD-1553B

1. This Military Standard is approved for use by all Department and Agencies of the Department of Defense.

2. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: Aeronautical Systems Division, Attn: ENAI, Wright-Patterson Air Force Base 45433, by using the self addressed Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

FOREWORD

This standard contains requirements for aircraft internal time division command/response multiplex data bus techniques which will be utilized in systems integration of aircraft subsystems. Even with the use of this standard, subtle differences will exist between multiplex data buses used on different aircraft due to particular aircraft mission requirements and the designer options allowed in this standard. The system designer must recognize this fact and design the multiplex bus controller hardware and software to accommodate such differences. These designer selected options must exist, so as to allow the necessary flexibility in the design of specific multiplex systems in order to provide for the control mechanism, architecture redundancy, degradation concept and traffic patterns peculiar to the specific aircraft mission requirements.

CONTENTS

CONTENTS (Cont'd)

CONTENTS (Cont'd)

CONTENTS (Cont'd)

Paragraph		Page

FIGURES

TABLES

APPENDIX

APPENDIX FIGURES

1. SCOPE

1.1 Scope. This standard establishes requirements for digital, command/response, time division multiplexing (Data Bus) techniques on aircraft. It encompasses the data bus line and its interface electronics illustrated on figure 1, and also defines the concept of operation and information flow on the multiplex data bus and the electrical and functional formats to be employed.

1.2 Application. When invoked in a specification or statement of work, these requirements shall apply to the multiplex data bus and associated equipment which is developed either alone or as a portion of an aircraft weapon system or subsystem development. The contractor is responsible for invoking all the applicable requirements of this Military Standard on any and all subcontractors he may employ.

2. REFERENCED DOCUMENTS

2.1 Issue of document. The following document, of the issue in effect on date of invitation for bid or request for proposal, forms a part of the standard to the extent specified herein.

SPECIFICATION

MILITARY

MIL-E-6051 Electromagnetic Compatibility Requirements, Systems

(Copies of specifications, standards, drawings, and publications required by contractors in connection with specific procurement functions should be obtained from the procuring activity or as directed by the contracting officer.)

3. DEFINITIONS

3.1 Bit. Contraction of binary digit: may be either zero or one. In information theory a binary digit is equal to one binary decision or the designation of one of two possible values or states of anything used to store or convey information.

3.2 Bit rate. The number of bits transmitted per second.

3.3 Pulse code modulation (PCM). The form of modulation in which the modulation signal is sampled, quantized, and coded so that each element of information consists of different types or numbers of pulses and spaces.

3.4 Time division multiplexing (TDM). The transmission of information from several signal sources through one communication system with different signal samples staggered in time to form a composite pulse train.

3.5 Half duplex. Operation of a data transfer system in either direction over a single line, but not in both directions on that line simultaneously.

3.6 Word. In this document a word is a sequence of 16 bits plus sync and parity. There are three types of words: command, status and data.

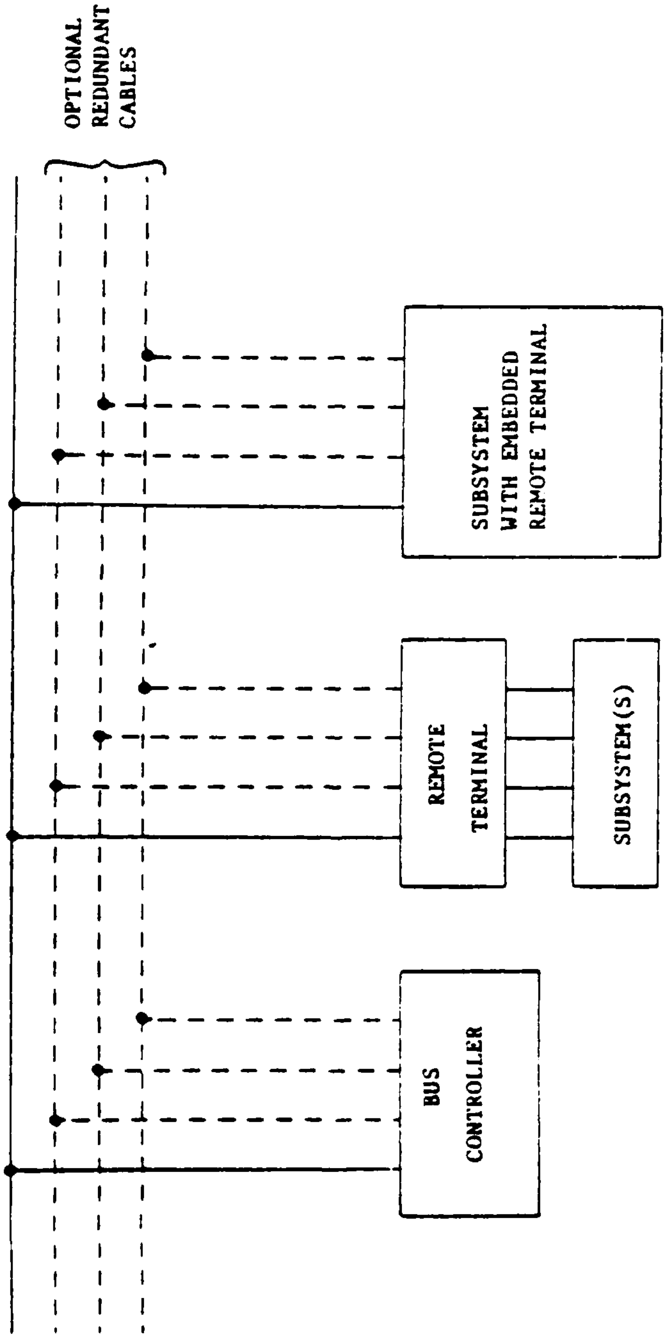


FIGURE 1. Sample multiplex data bus architecture.

3.7 Message. A single message is the transmission of a command word, status word, and data words if they are specified. For the case of a remote terminal to remote terminal (RT to RT) transmission, the message shall include the two command words, the two status words, and data words.

3.8 Subsystem. The device or functional unit receiving data transfer service from the data bus.

3.9 Data bus. Whenever a data bus or bus is referred to in this document it shall imply all the hardware including twisted shielded pair cables, isolation resistors, transformers, etc., required to provide a single data path between the bus controller and all the associated remote terminals.

3.10 Terminal. The electronic module necessary to interface the data bus with the subsystem and the subsystem with the data bus. Terminals may exist as separate line replaceable units (LRU's) or be contained within the elements of the subsystem.

3.11 Bus controller. The terminal assigned the task of initiating information transfers on the data bus.

3.12 Bus monitor. The terminal assigned the task of receiving bus traffic and extracting selected information to be used at a later time.

3.13 Remote terminal (RT). All terminals not operating as the bus controller or as a bus monitor.

3.14 Asynchronous operation. For the purpose of this standard, asynchronous operation is the use of an independent clock source in each terminal for message transmission. Decoding is achieved in receiving terminals using clock information derived from the message.

3.15 Dynamic bus control. The operation of a data bus system in which designated terminals are offered control of the data bus.

3.16 Command/Response. Operation of a data bus system such that remote terminals receive and transmit data only when commanded to do so by the bus controller.

3.17 Redundant data bus. The use of more than one data bus to provide more than one data path between the subsystems, i.e., dual redundant data bus, tri-redundant data bus, etc.

3.18 Broadcast. Operation of a data bus system such that information transmitted by the bus controller or a remote terminal is addressed to more than one of the remote terminals connected to the data bus.

3.19 Mode code. A means by which the bus controller can communicate with the multiplex bus related hardware, in order to assist in the management of information flow.

4. GENERAL REQUIREMENTS

4.1 Test and operating requirements. All requirements as specified herein shall be valid over the environmental conditions which the multiplex data bus system shall be required to operate.

4.2 Data bus operation. The multiplex data bus system in its most elemental configuration shall be as shown on figure 1. The multiplex data bus system shall function asynchronously in a command/response mode, and transmission shall occur in a half-duplex manner. Sole control of information transmission on the bus shall reside with the bus controller, which shall initiate all transmissions. The information flow on the data bus shall be comprised of messages which are, in turn, formed by three types of words (command, data, and status) as defined in 4.3.3.5.

4.3 Characteristics

4.3.1 Data form. Digital data may be transmitted in any desired form, provided that the chosen form shall be compatible with the message and word formats defined in this standard. Any unused bit positions in a word shall be transmitted as logic zeros.

4.3.2 Bit priority. The most significant bit shall be transmitted first with the less significant bits following in descending order of value in the data word. The number of bits required to define a quantity shall be consistent with the resolution or accuracy required. In the event that multiple precision quantities (information accuracy or resolution requiring more than 16 bits) are transmitted, the most significant bits shall be transmitted first, followed by the word(s) containing the lesser significant bits in numerical descending order. Bit packing of multiple quantities in a single data word is permitted.

4.3.3 Transmission method

4.3.3.1 Modulation. The signal shall be transferred over the data bus in serial digital pulse code modulation form.

4.3.3.2 Data code. The data code shall be Manchester II bi-phase level. A logic one shall be transmitted as a bipolar coded signal 1/0 (i.e., a positive pulse followed by a negative pulse). A logic zero shall be a bipolar coded signal 0/1 (i.e., a negative pulse followed by a positive pulse). A transition through zero occurs at the midpoint of each bit time (see figure 2).

4.3.3.3 Transmission bit rate. The transmission bit rate on the bus shall be 1.0 megabit per second with a combined accuracy and long-term stability of ± 0.1 percent (i.e., ± 1000 Hertz (Hz)). The short-term stability (i.e., stability over 1.0 second interval) shall be at least 0.01 percent (i.e., ± 100 Hz).

4.3.3.4 Word size. The word size shall be 16 bits plus the sync waveform and the parity bit for a total of 20 bits times as shown on figure 3.

4.3.3.5 Word formats. The word formats shall be as shown on figure 3 for the command, data, and status words.

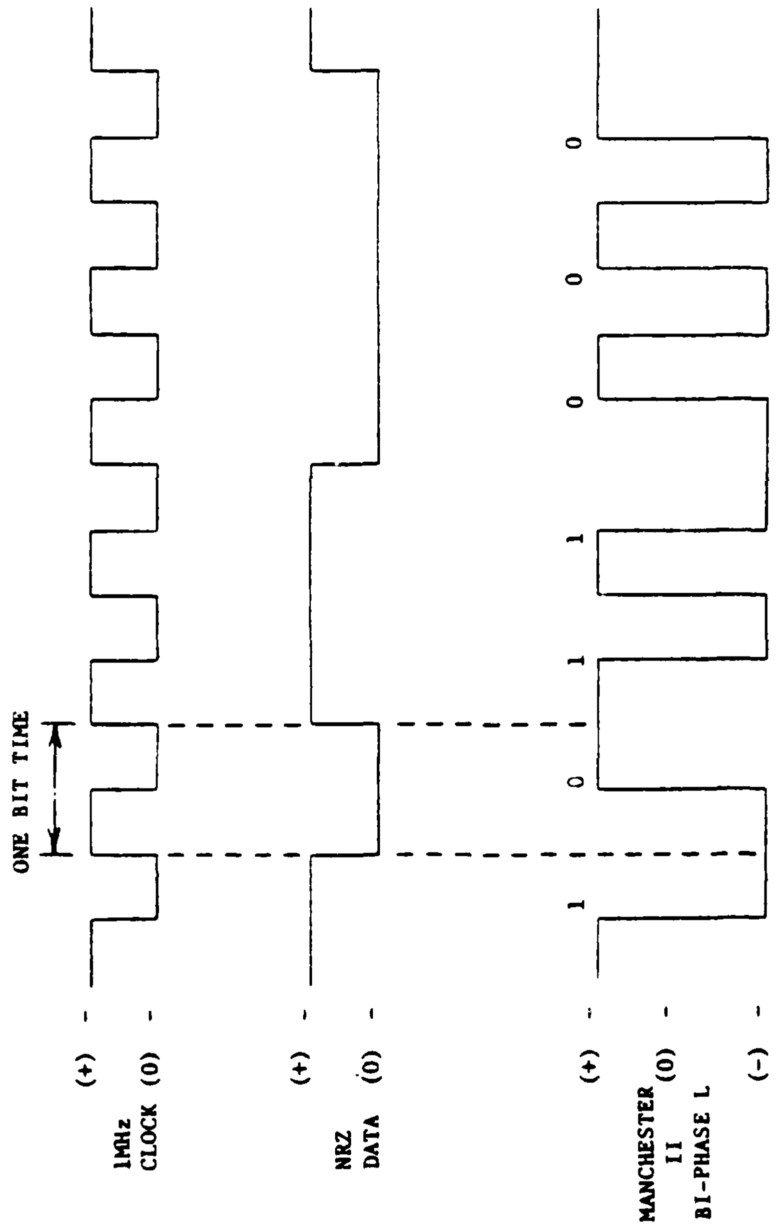
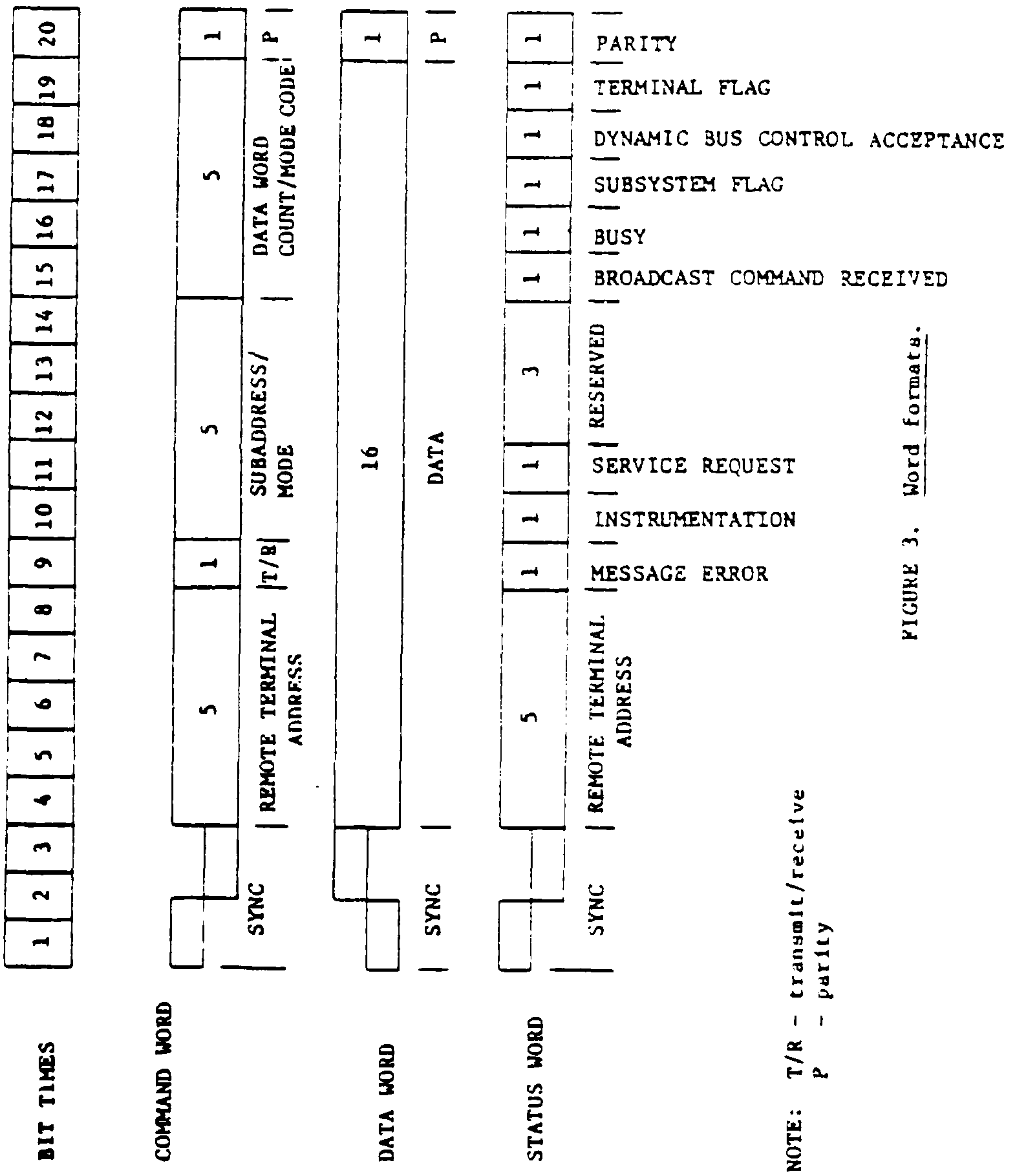
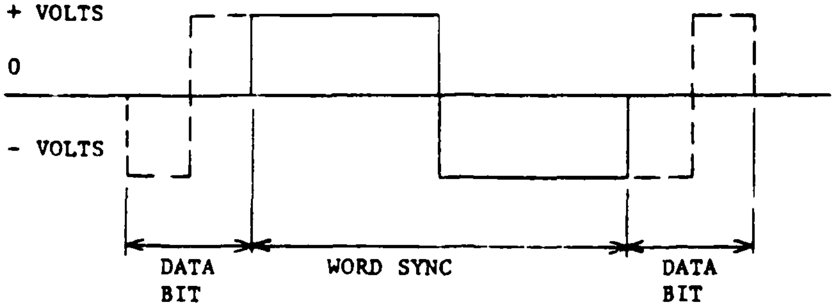


FIGURE 2. Data encoding.



NOTE: T/R - transmit/receive
P - parity

FIGURE 3. Word formats.



FIRURE 4. Command and status sync.

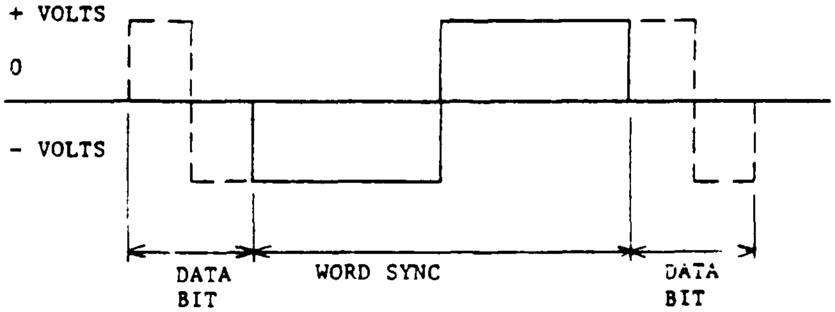


FIGURE 5. Data sync.

4.3.3.5.1 Command word. A command word shall be comprised of a sync waveform, remote terminal address field, transmit/receive (T/R) bit, subaddress/mode field, word count/mode code field, and a parity (P) bit (see figure 3).

4.3.3.5.1.1 Sync. The command sync waveform shall be an invalid Manchester waveform as shown on figure 4. The width shall be three bit times, with the sync waveform being positive for the first one and one-half bit times, and then negative for the following one and one-half bit times. If the next bit following the sync waveform is a logic zero, then the last half of the sync waveform will have an apparent width of two clock periods due to the Manchester encoding.

4.3.3.5.1.2 Remote terminal address. The next five bits following the sync shall be the RT address. Each RT shall be assigned a unique address. Decimal address 31 (11111) shall not be assigned as a unique address. In addition to its unique address, a RT shall be assigned decimal address 31 (11111) as the common address, if the broadcast option is used.

4.3.3.5.1.3 Transmit/receive. The next bit following the remote terminal address shall be the T/R bit, which shall indicate the action required of the RT. A logic zero shall indicate the RT is to receive, and a logic one shall indicate the RT is to transmit.

4.3.3.5.1.4 Subaddress/mode. The next five bits following the R/T bit shall be utilized to indicate an RT subaddress or use of mode control, as is dictated by the individual terminal requirements. The subaddress/mode values of 00000 and 11111 are reserved for special purposes, as specified in 4.3.3.5.1.7, and shall not be utilized for any other function.

4.3.3.5.1.5 Data word count/mode code. The next five bits following the subaddress/mode field shall be the quantity of data words to be either sent out or received by the RT or the optional mode code as specified in 4.3.3.5.1.7. A maximum of 32 data words may be transmitted or received in any one message block. All 1's shall indicate a decimal count of 31, and all 0's shall indicate a decimal count of 32.

4.3.3.5.1.6 Parity. The last bit in the word shall be used for parity over the preceding 16 bits. Odd parity shall be utilized.

4.3.3.5.1.7 Optional mode control. For RT's exercising this option a subaddress/mode code of 00000 or 11111 shall imply that the contents of the data word count/mode code field are to be decoded as a five bit mode command. The mode code shall only be used to communicate with the multiplex bus related hardware, and to assist in the management of information flow, and not to extract data from or feed data to a functional subsystem. Codes 00000 through 01111 shall only be used for mode codes which do not require transfer of a data word. For these codes, the T/R bit shall be set to 1. Codes 10000 through 11111 shall only be used for mode codes which require transfer of a single data word. For these mode codes, the T/R bit shall indicate the direction of data word flow as specified in 4.3.3.5.1.3. No multiple data word transfer shall be implemented with any mode code. The mode codes are reserved for the specific functions as specified in table I and shall not be used for any other purpose. If the designer chooses to implement any of these functions, the specific

codes, T/R bit assignments, and use of a data word, shall be used as indicated. The use of the broadcast command option shall only be applied to particular mode codes as specified in table I.

4.3.3.5.1.7.1 Dynamic bus control. The controller shall issue a transmit command to an RT capable of performing the bus control function. This RT shall respond with a status word as specified in 4.3.3.5.3. Control of the data bus passes from the offering bus controller to the accepting RT upon completion of the transmission of the status word by the RT. If the RT rejects control of the data bus, the offering bus controller retains control of the data bus.

4.3.3.5.1.7.2 Synchronize (without data word). This command shall cause the RT to synchronize (e.g., to reset the internal timer, to start a sequence, etc.). The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.3 Transmit status word. This command shall cause the RT to transmit the status word associated with the last valid command word preceding this command. This mode command shall not alter the state of the status word.

4.3.3.5.1.7.4 Initiate self test. This command shall be used to initiate self test within the RT. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.5 Transmitter shutdown. This command (to only be used with dual redundant bus systems) shall cause the RT to disable the transmitter associated with the redundant bus. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.

4.3.3.5.1.7.6 Override transmitter shutdown. This command (to only be used with dual redundant bus system) shall cause the RT to enable a transmitter which was previously disabled. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3 after this command.

4.3.3.5.1.7.7 Inhibit terminal flag (T/F) bit. This command shall cause the RT to set the T/F bit in the status word specified in 4.3.3.5.3 to logic zero until otherwise commanded. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.8 Override inhibit T/F bit. This command shall cause the RT to override the inhibit T/F bit specified in 4.3.3.5.1.7.7. The RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.9 Reset remote terminal. This command shall be used to reset the RT to a power up initialized state. The RT shall first transmit its status word, and then reset.

4.3.3.5.1.7.10 Reserved mode codes (01001 to 01111). These mode codes are reserved for future use and shall not be used.

TABLE I. Assigned mode codes

T/R Bit	Mode Code	Function	Associated Data Word	Broadcast Command Allowed
1	00000	Dynamic Bus Control	No	No
1	00001	Synchronize	No	Yes
1	00010	Transmit Status Word	No	No
1	00011	Initiate Self Test	No	Yes
1	00100	Transmitter Shutdown	No	Yes
1	00101	Override Transmitter Shutdown	No	Yes
1	00110	Inhibit Terminal Flag Bit	No	Yes
1	00111	Override Inhibit Terminal Flag Bit	No	Yes
1	01000	Reset Remote Terminal	No	Yes
1	01001	Reserved	No	TBD
	↓	↓	↓	↓
1	01111	Reserved	No	TBD
1	10000	Transmit Vector Word	Yes	No
0	10001	Synchronize	Yes	Yes
1	10010	Transmit Last Command	Yes	No
1	10011	Transmit BIT Word	Yes	No
0	10100	Selected Transmitter Shutdown	Yes	Yes
0	10101	Override Selected Transmitter Shutdown	Yes	Yes
1 or 0	10110	Reserved	Yes	TBD
	↓	↓	↓	↓
1 or 0	11111	Reserved	Yes	TBD

NOTE: To be determined (TBD)

4.3.3.5.1.7.11 Transmit vector word. This command shall cause the RT to transmit a status word as specified in 4.3.3.5.3 and a data word containing service request information.

4.3.3.5.1.7.12 Synchronize (with data word). The RT shall receive a command word followed by a data word as specified in 4.3.3.5.2. The data word shall contain synchronization information for the RT. After receiving the command and data word, the RT shall transmit the status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.13 Transmit last command word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word which contains bits 4-19 of the last command word, excluding a transmit last command word mode code received by the RT. This mode command shall not alter the state of the RT's status word.

4.3.3.5.1.7.14 Transmit built-in-test (BIT) word. This command shall cause the RT to transmit its status word as specified in 4.3.3.5.3 followed by a single data word containing the RT BIT data. This function is intended to supplement the available bits in the status word when the RT hardware is sufficiently complex to warrant its use. The data word, containing the RT BIT data, shall not be altered by the reception of a transmit last command or a transmit status word mode code. This function shall not be used to convey BIT data from the associated subsystem(s).

4.3.3.5.1.7.15 Selected transmitter shutdown. This command shall cause the RT to disable the transmitter associated with a specified redundant data bus. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be disabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to shut down a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.16 Override selected transmitter shutdown. This command shall cause the RT to enable a transmitter which was previously disabled. The command is designed for use with systems employing more than two redundant buses. The transmitter that is to be enabled shall be identified in the data word following the command word in the format as specified in 4.3.3.5.2. The RT shall not comply with a command to enable a transmitter on the bus from which this command is received. In all cases, the RT shall respond with a status word as specified in 4.3.3.5.3.

4.3.3.5.1.7.17 Reserved mode codes (10110 to 11111). These mode codes are reserved for future use and shall not be used.

4.3.3.5.2 Data word. A data word shall be comprised of a sync waveform, data bits, and a parity bit (see figure 3).

4.3.3.5.2.1 Sync. The data sync waveform shall be an invalid Manchester waveform as shown on figure 5. The width shall be three bit times, with the waveform being negative for the first one and one-half bit times, and then positive for the following one and one-half bit times. Note that if the bits preceding and following the sync are logic ones, then the apparent width of the sync waveform will be increased to four bit times.

4.3.3.5.2.2 Data. The sixteen bits following the sync shall be utilized for data transmission as specified in 4.3.2.

4.3.3.5.2.3 Parity. The last bit shall be utilized for parity as specified in 4.3.3.5.1.6.

4.3.3.5.3 Status word. A status word shall be comprised of a sync waveform, RT address, message error bit, instrumentation bit, service request bit, three reserved bits, broadcast command received bit, busy bit, subsystem flag bit, dynamic bus control acceptance bit, terminal flag bit, and a parity bit. For optional broadcast operation, transmission of the status word shall be suppressed as specified in 4.3.3.6.7.

4.3.3.5.3.1 Sync. The status sync waveform shall be as specified in 4.3.3.5.1.1.

4.3.3.5.3.2 RT address. The next five bits following the sync shall contain the address of the RT which is transmitting the status word as defined in 4.3.3.5.1.2.

4.3.3.5.3.3 Message error bit. The status word bit at bit time nine (see figure 3) shall be utilized to indicate that one or more of the data words associated with the preceding receive command word from the bus controller has failed to pass the RT's validity tests as specified in 4.4.1.1. This bit shall also be set under the conditions specified in 4.4.1.2, 4.4.3.4 and 4.4.3.6. A logic one shall indicate the presence of a message error, and a logic zero shall show its absence. All RT's shall implement the message error bit.

4.3.3.5.3.4 Instrumentation bit. The status word at bit time ten (see figure 3) shall be reserved for the instrumentation bit and shall always be a logic zero. This bit is intended to be used in conjunction with a logic one in bit time ten of the command word to distinguish between a command word and a status word. The use of the instrumentation bit is optional.

4.3.3.5.3.5 Service request bit. The status word bit at bit time eleven (see figure 3) shall be reserved for the service request bit. The use of this bit is optional. This bit when used, shall indicate the need for the bus controller to take specific predefined actions relative to either the RT or associated subsystem. Multiple subsystems, interfaced to a single RT, which individually require a service request signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific requesting subsystem. The service request bit is intended to be used only to trigger data transfer operations which take place on an exception rather than periodic basis. A logic one shall indicate the presence of a service request, and a logic zero its absence. If this function is not implemented, the bit shall be set to zero.

4.3.3.5.3.6 Reserved status bits. The status word bits at bit times twelve through fourteen are reserved for future use and shall not be used. These bits shall be set to a logic zero.

4.3.3.5.3.7 Broadcast command received bit. The status word at bit time fifteen shall be set to a logic one to indicate that the preceding valid command word was a broadcast command and a logic zero shall show it was not a broadcast command. If the broadcast command option is not used, this bit shall be set to a logic zero.

4.3.3.5.3.8 Busy bit. The status word bit at bit time sixteen (see figure 3) shall be reserved for the busy bit. The use of this bit is optional. This bit, when used, shall indicate that the RT or subsystem is unable to move data to or from the subsystem in compliance with the bus controller's command. A logic one shall indicate the presence of a busy condition, and a logic zero its absence. In the event the busy bit is set in response to a transmit command, then the RT shall transmit its status word only. If this function is not implemented, the bit shall be set to logic zero.

4.3.3.5.3.9 Subsystem flag bit. The status word bit at bit time seventeen (see figure 3) shall be reserved for the subsystem flag bit. The use of this bit is optional. This bit, when used, shall flag a subsystem fault condition, and alert the bus controller to potentially invalid data. Multiple subsystems, interfaced to a single RT, which individually require a subsystem flag bit signal shall logically OR their individual signals into the single status word bit. In the event this logical OR is performed, then the designer must make provisions in a separate data word to identify the specific reporting subsystem. A logic one shall indicate the presence of the flag, and a logic zero its absence. If not used, this bit shall be set to logic zero.

4.3.3.5.3.10 Dynamic bus control acceptance bit. The status word bit at bit time eighteen (see figure 3) shall be reserved for the acceptance of dynamic bus control. This bit shall be used if the RT implements the optional dynamic bus control function. This bit, when used, shall indicate acceptance or rejection of a dynamic bus control offer as specified in 4.3.3.5.1.7.1. A logic one shall indicate acceptance of control, and a logic zero shall indicate rejection of control. If this function is not used, this bit shall be set to logic zero.

4.3.3.5.3.11 Terminal flag bit. The status word bit at bit time nineteen (see figure 3) shall be reserved for the terminal flag function. The use of this bit is optional. This bit, when used, shall flag a RT fault condition. A logic one shall indicate the presence of the flag, and a logic zero, its absence. If not used, this bit shall be set to logic zero.

4.3.3.5.3.12 Parity bit. The least significant bit in the status word shall be utilized for parity as specified in 4.3.3.5.1.6.

4.3.3.5.4 Status word reset. The status word bit, with the exception of the address, shall be set to logic zero after a valid command word is received by the RT with the exception as specified in 4.3.3.5.1.7. If the conditions which caused bits in the status word to be set (e.g., terminal flag) continue after the bits are reset to logic zero, then the affected status word bit shall be again set, and then transmitted on the bus as required.

4.3.3.6 Message formats. The messages transmitted on the data bus shall be in accordance with the formats on figure 6 and figure 7. The maximum and minimum response times shall be as stated in 4.3.3.7 and 4.3.3.8. No message formats, other than those defined herein, shall be used on the bus.

4.3.3.6.1 Bus controller to remote terminal transfers. The bus controller shall issue a receive command followed by the specified number of data words. The RT shall, after message validation, transmit a status word back to the controller. The command and data words shall be transmitted in a contiguous fashion with no interword gaps.

4.3.3.6.2 Remote terminal to bus controller transfers. The bus controller shall issue a transmit command to the RT. The RT shall, after command word validation, transmit a status word back to the bus controller, followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no interword gaps.

4.3.3.6.3 Remote terminal to remote terminal transfers. The bus controller shall issue a receive command to RT A followed contiguously by a transmit command to RT B. RT B shall, after command validation, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. At the conclusion of the data transmission by RT B, RT A shall transmit a status word within the specified time period.

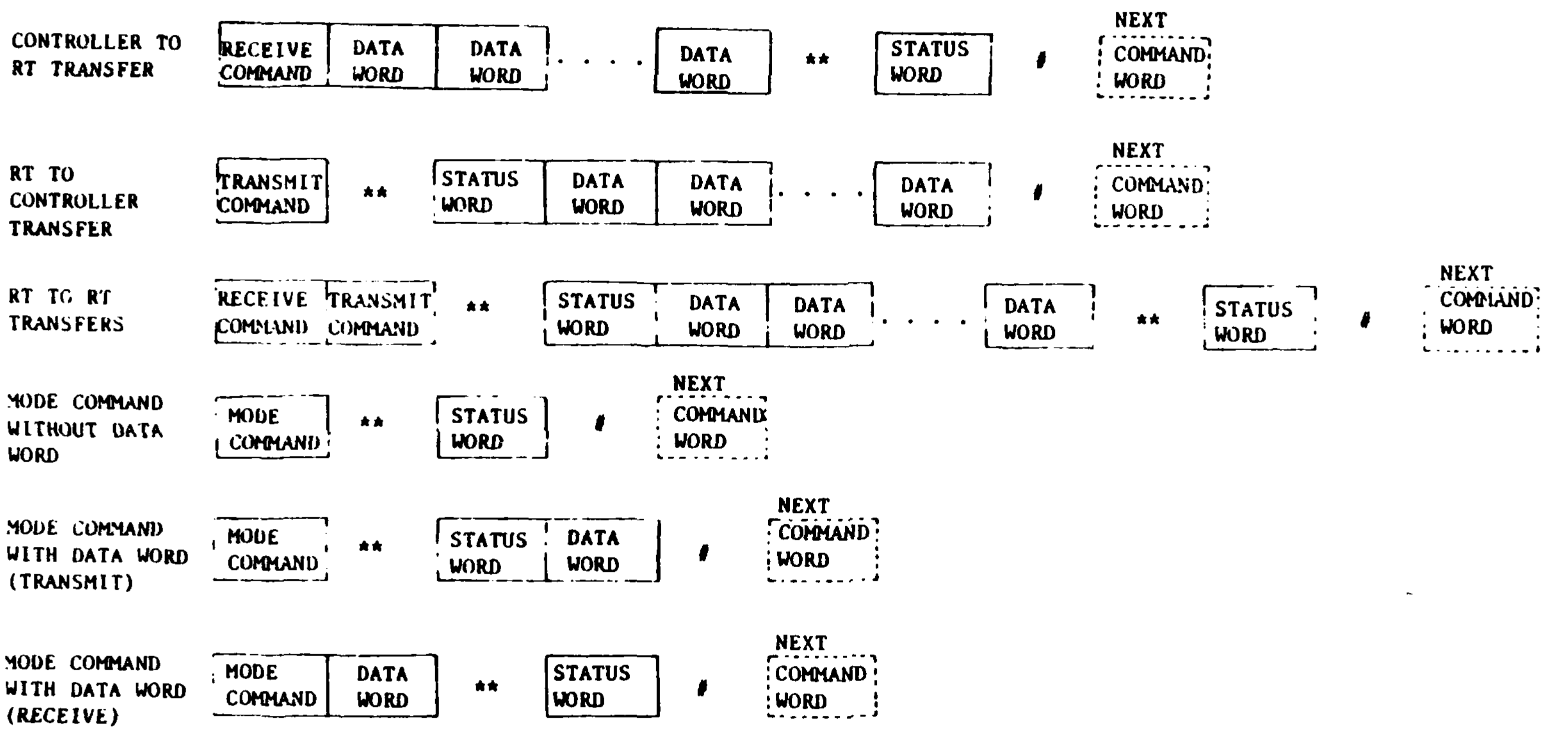
4.3.3.6.4 Mode command without data word. The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word.

4.3.3.6.5 Mode command with data word (transmit). The bus controller shall issue a transmit command to the RT using a mode code specified in table I. The RT shall, after command word validation, transmit a status word followed by one data word. The status word and data word shall be transmitted in a contiguous fashion with no gap.

4.3.3.6.6 Mode command with data word (receive). The bus controller shall issue a receive command to the RT using a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT shall, after command and data word validation, transmit a status word back to the controller.

4.3.3.6.7 Optional broadcast command. See 10.6 for additional information on the use of the broadcast command.

4.3.3.6.7.1 Bus controller to remote terminal(s) transfer (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field followed by the specified number of data words. The command word and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message validation, set the broadcast command received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.



NOTE: # INTERMESSAGE GAP
** RESPONSE TIME

FIGURE 6. <u>Information transfer formats.</u>

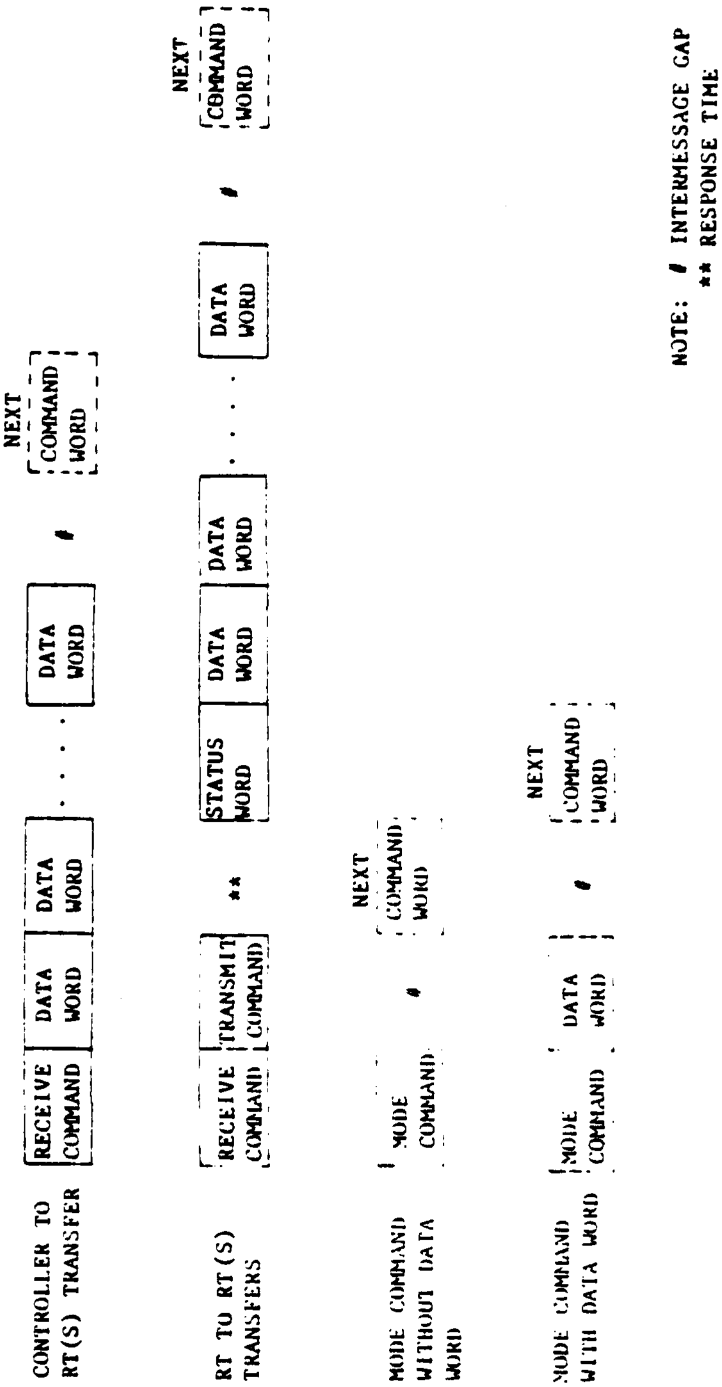


FIGURE 7. Broadcast information transfer formats.

4.3.3.6.7.2 Remote terminal to remote terminal(s) transfers (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field followed by a transmit command to RT A using the RT's address. RT A shall, after command word validation, transmit a status word followed by the specified number of data words. The status and data words shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option, excluding RT A, shall after message validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.3 Mode command without data word (broadcast). The bus controller shall issue a transmit command word with 11111 in the RT address field, and a mode code specified in table I. The RT(s) with the broadcast option shall after command word validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.6.7.4 Mode command with data word (broadcast). The bus controller shall issue a receive command word with 11111 in the RT address field and a mode code specified in table I, followed by one data word. The command word and data word shall be transmitted in a contiguous fashion with no gap. The RT(s) with the broadcast option shall after message validation, set the broadcast received bit in the status word as specified in 4.3.3.5.3.7 and shall not transmit the status word.

4.3.3.7 Intermessage gap. The bus controller shall provide a minimum gap time of 4.0 microseconds (µs) between messages as shown on figure 6 and figure 7. This time period, shown as T on figure 8, is measured at point A of the bus controller as shown on figure 9 or figure 10. The time is measured from the mid-bit zero crossing of the last bit of the preceding message to mid-zero crossing of the next command word sync.

4.3.3.8 Response time. The RT shall respond, in accordance with 4.3.3.6, to a valid command word within the time period of 4.0 to 12.0 µs. This time period, shown as T on figure 8, is measured at point A of the RT as shown on figure 9 or figure 10. The time is measured from the mid bit zero crossing of the last word as specified in 4.3.3.6 and as shown on figure 6 and figure 7 to the mid-zero crossing of the status word sync.

4.3.3.9 Minimum no-response time-out. The minimum time that a terminal shall wait before considering that a response as specified in 4.3.3.8 has not occurred shall be 14.0 µs. The time is measured from the mid-bit zero crossing of the last bit of the last word to the mid-zero crossing of the expected status word sync at point A of the terminal as shown on figure 9 or figure 10.

4.4 Terminal operation.

4.4.1 Common operation. Terminals shall have common operating capabilities as specified in the following paragraphs.

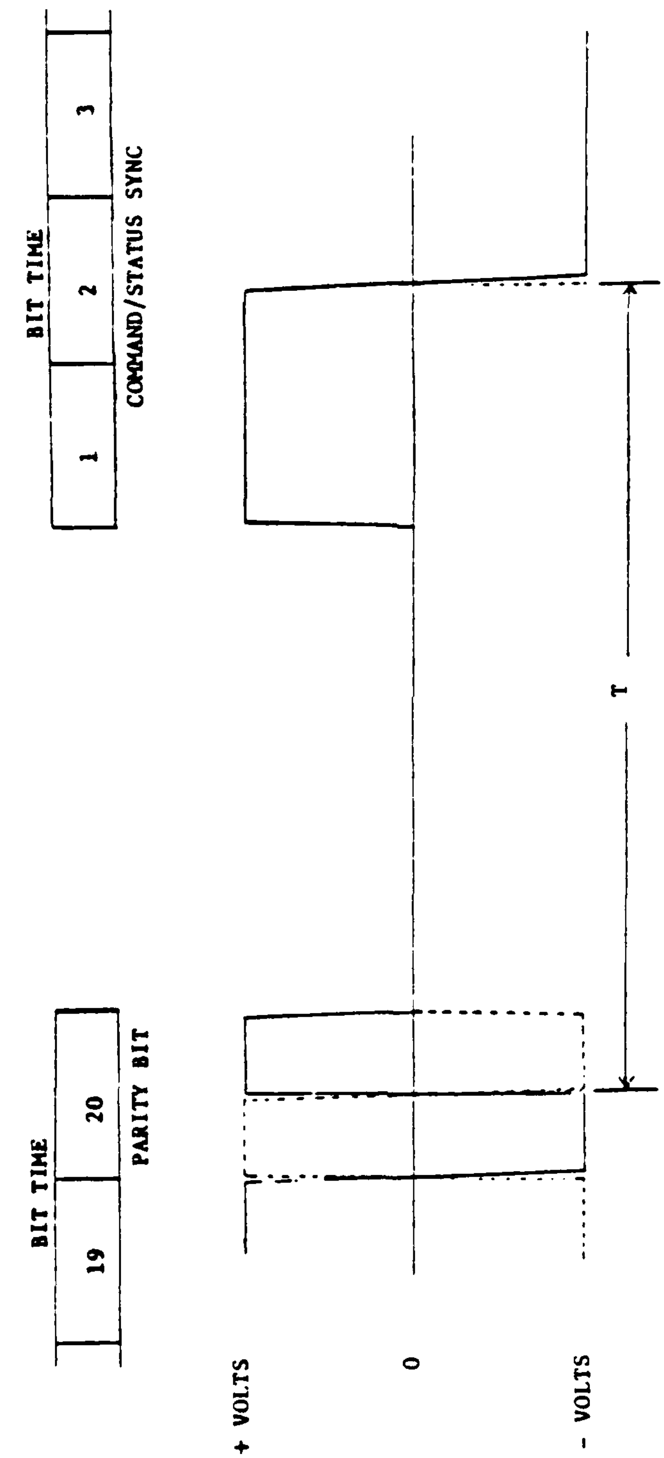


FIGURE 8. Intermessage gap and response time.

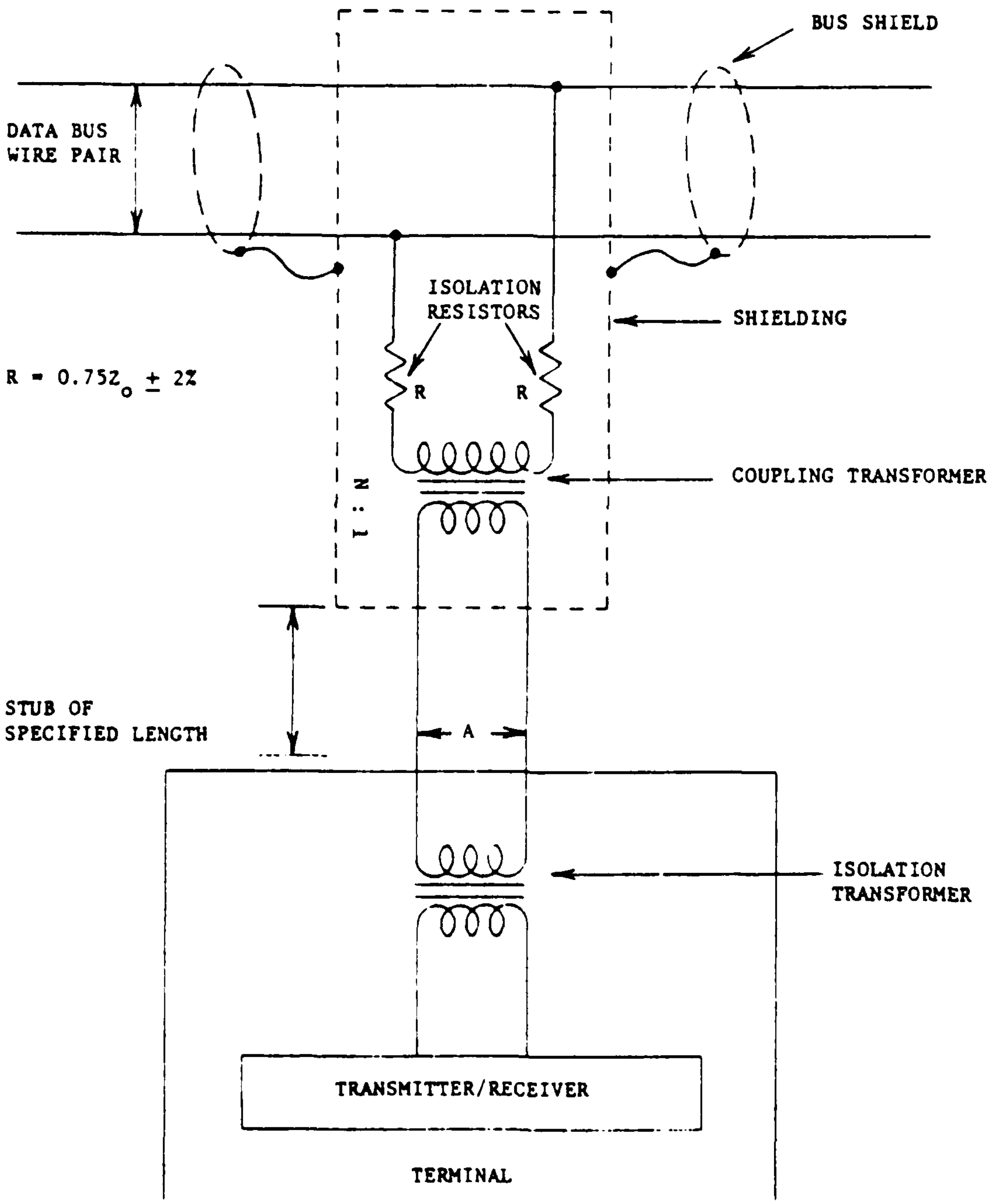


FIGURE 9. Data bus interface using transformer coupling.

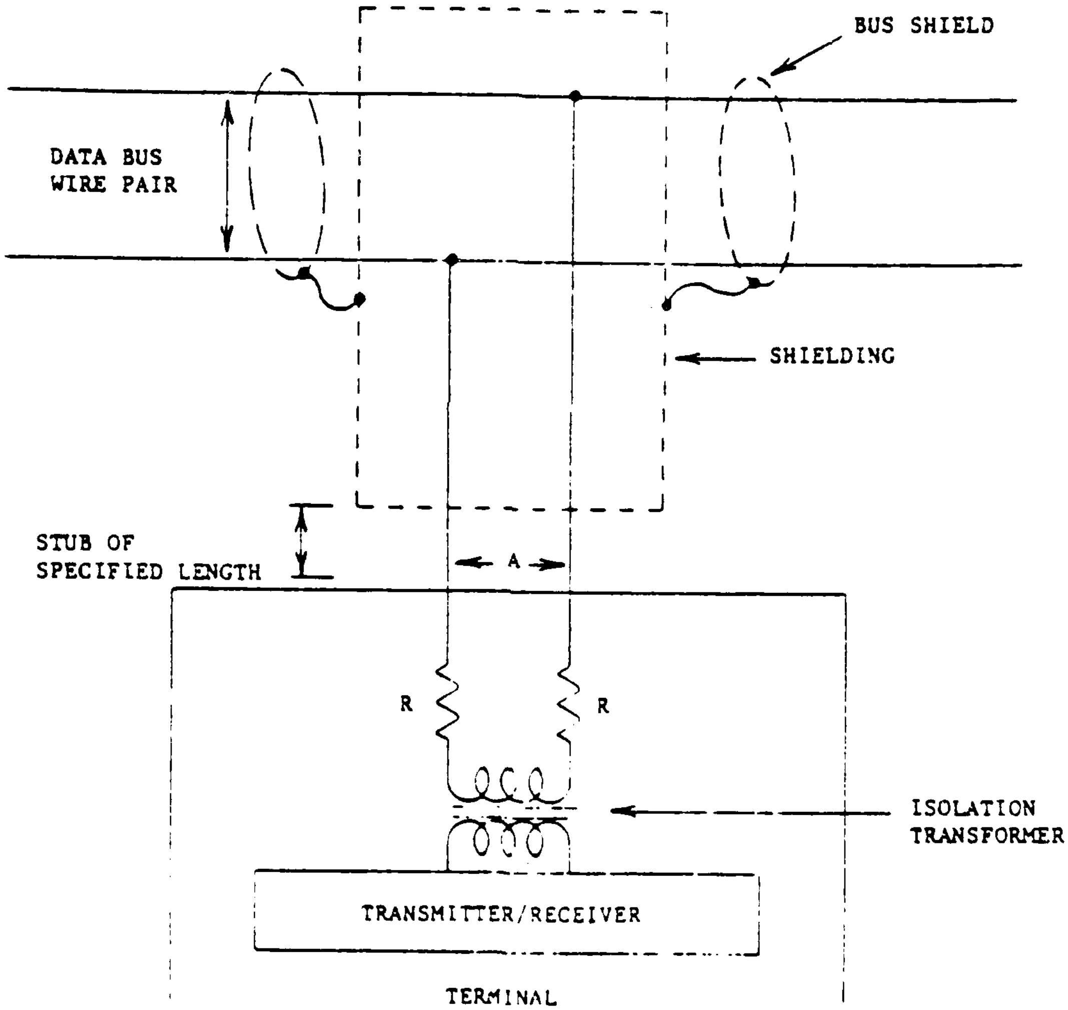


FIGURE 10. Data bus interface using direct coupling.

4.4.1.1 Word validation. The terminal shall insure that each word conforms to the following minimum criteria:

a. The word begins with a valid sync field.
b. The bits are in a valid Manchester II code.
c. The information field has 16 bits plus parity.
d. The word parity is odd.

When a word fails to conform to the preceding criteria, the word shall be considered invalid.

4.4.1.2 Transmission continuity. The terminal shall verify that the message is contiguous as defined in 4.3.3.6. Improperly timed data syncs shall be considered a message error.

4.4.1.3 Terminal fail-safe. The terminal shall contain a hardware implemented time-out to preclude a signal transmission of greater than 800.0 µs. This hardware shall not preclude a correct transmission in response to a command. Reset of this time-out function shall be performed by the reception of a valid command on the bus on which the time-out has occurred.

4.4.2 Bus controller operation. A terminal operating as a bus controller shall be responsible for sending data bus commands, participating in data transfers, receiving status responses, and monitoring system status as defined in this standard. the bus controller function may be embodied as either a stand-alone terminal, whose sole function is to control the data bus(s), or contained within a subsystem. Only one terminal shall be in active control of a data bus at any one time.

4.4.3 Remote terminal.

4.4.3.1 Operation. A remote terminal (RT) shall operate in response to valid commands received from the bus controller. The RT shall accept a command word as valid when the command word meets the criteria of 4.4.1.1, and the command word contains a terminal address which matches the RT address or an address of 11111, if the RT has the broadcast option.

4.4.3.2 Superseding valid commands. The RT shall be capable of receiving a command word on the data bus after the minimum intermessage gap time as specified in 4.3.3.7 has been exceeded, when the RT is not in the time period T as specified in 4.3.3.8 prior to the transmission of a status word, and when it is not transmitting on that data bus. A second valid command word sent to an RT shall take precedence over the previous command. The RT shall respond to the second valid command as specified in 4.3.3.8.

4.4.3.3 Invalid commands. A remote terminal shall not respond to a command word which fails to meet the criteria specified in 4.4.3.1.

4.4.3.4 Illegal command. An illegal command is a valid command as specified in 4.4.3.1, where the bits in the subaddress/mode field, data word count/mode code field, and the T/R bit indicate a mode command, subaddress, or word count that has not been implemented in the RT. It is the responsibility of the bus controller to assure that no illegal commands are sent out. The RT designer has the option of monitoring for illegal commands. If an RT that is designed with this option detects an illegal command and the proper number of contiguous

valid data words as specified by the illegal command word, it shall respond with a status word only, setting the message error bit, and not use the information received.

4.4.3.5 Valid data reception. The remote terminal shall respond with a status word when a valid command word and the proper number of contiguous valid data words are received, or a single valid word associated with a mode code is received. Each data word shall meet the criteria specified in 4.4.1.1.

4.4.3.6 Invalid data reception. Any data word(s) associated with a valid receive command that does not meet the criteria specified in 4.4.1.1 and 4.4.1.2 or an error in the data word count shall cause the remote terminal to set the message error bit in the status word to a logic one and suppress the transmission of the status word. If a message error has occurred, then the entire message shall be considered invalid.

4.4.4 Bus monitor operation. A terminal operating as a bus monitor shall receive bus traffic and extract selected information. While operating as a bus monitor, the terminal shall not respond to any message except one containing its own unique address if one is assigned. All information obtained while acting as a bus monitor shall be strictly used for off-line applications (e.g., flight test recording, maintenance recording or mission analysis) or to provide the back-up bus controller sufficient information to take over as the bus controller.

4.5 Hardware characteristics.

4.5.1 Data bus characteristics.

4.5.1.1 Cable. The cable used for the main bus and all stubs shall be a two conductor, twisted, shielded, jacketed cable. The wire-to-wire distributed capacitance shall not exceed 30.0 picofarads per foot. The cables shall be formed with not less than four twists per foot where a twist is defined as a 360 degree rotation of the wire pairs; and, the cable shield shall provide a minimum of 75.0 percent coverage.

4.5.1.2 Characteristic impedance. The nominal characteristic impedance of the cable (Z_0) shall be within the range of 70.0 ohms to 85.0 ohms at a sinusoidal frequency of 1.0 megahertz (MHz).

4.5.1.3 Cable attenuation. At the frequency of 4.5.1.2, the cable power loss shall not exceed 1.5 decibels (dB)/100 feet (ft).

4.5.1.4 Cable termination. The two ends of the cable shall be terminated with a resistance, equal to the selected cable nominal characteristic impedance (Z_0) ± 2.0 percent.

4.5.1.5 Cable stub requirements. The cable shall be coupled to the terminal as shown on figure 9 or figure 10. The use of long stubs is discouraged, and the length of a stub should be minimized. However, if installation requirements dictate, stub lengths exceeding those lengths specified in 4.5.1.5.1 and 4.5.1.5.2 are permissible.

4.5.1.5.1 Transformer coupled stubs. The length of a transformer coupled stub should not exceed 20 feet. If a transformer coupled stub is used, then the following shall apply.

4.5.1.5.1.1 Coupling transformer. A coupling transformer, as shown on figure 9, shall be required. This transformer shall have a turns ratio of 1:1.41 ± 3.0 percent, with the higher turns on the isolation resistor side of the stub.

4.5.1.5.1.1.1 Transformer input impedance. The open circuit impedance as seen at point B on figure 11 shall be greater than 3000 ohms over the frequency range of 75.0 kilohertz (kHz) to 1.0 megahertz (MHz), when measured with a 1.0 V root-mean-square (RMS) sin wave.

4.5.1.5.1.1.2 Transformer waveform integrity. The droop of the transformer using the test configuration shown on figure 11 at point B, shall not exceed 20.0 percent. Overshoot and ringing as measured at point B shall be less that ± 1.0 V peak. For this test, R shall equal 360.0 ohms ± 5.0 percent and the input A of figure 11 shall be a 250.0 kHz square wave. 27.0 V peak-to-peak, with a rise and fall time no greater than 100 nanoseconds (ns).

4.5.1.5.1.1.3 Transformer common mode rejection. The coupling transformer shall have a common mode rejection ratio greater than 45.0 dB at 1.0 MHz.

4.5.1.5.1.2 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 0.75 Z_o ohms plus or minus 2.0 percent, where Z_o is the selected cable nominal characteristic impedance. The impedance placed across the data bus cable shall be no less than 1.5 Z_o ohms for any failure of the coupling transformer, cable stub, or terminal transmitter/receiver.

4.5.1.5.1.3 Cable coupling. All coupling transformers and isolation resistors, as specified in 4.5.1.5.1.1 and 4.5.1.5.1.2, shall have continuous shielding which will provide a minimum of 75 percent coverage. The isolation resistors and coupling transformers shall be placed at minimum possible distance from the junction of the stub to the main bus.

4.5.1.5.1.4 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 9 shall have a peak-to-peak amplitude, line-to-line within the range of 1.0 and 14.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault impedance specified in 4.5.1.5.1.2 on the data bus. This shall also include the worse case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.2 Direct coupled stubs. The length of a direct coupled stub should not exceed 1 foot. Refer to 10.5 for comments concerning direct coupled stubs. If a direct coupled stub is used, then the following shall apply.

4.5.1.5.2.1 Fault isolation. An isolation resistor shall be placed in series with each connection to the data bus cable. This resistor shall have a value of 55.0 ohms plus or minus 2.0 percent. The isolation resistors shall be placed within the RT as shown on figure 10.

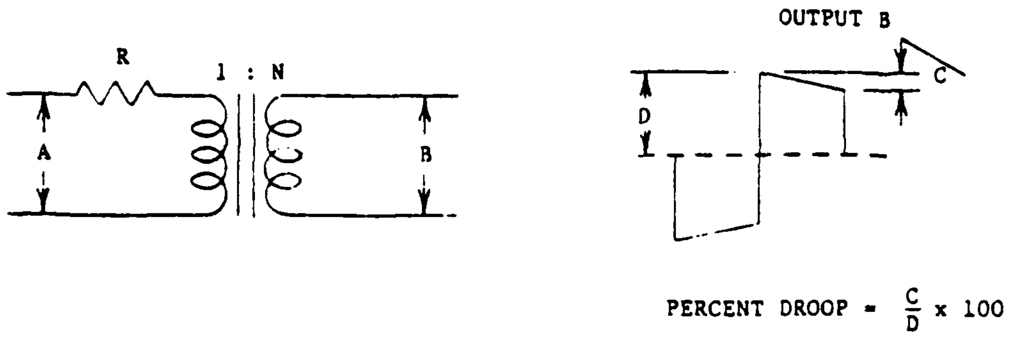


FIGURE 11. Coupling transformer.

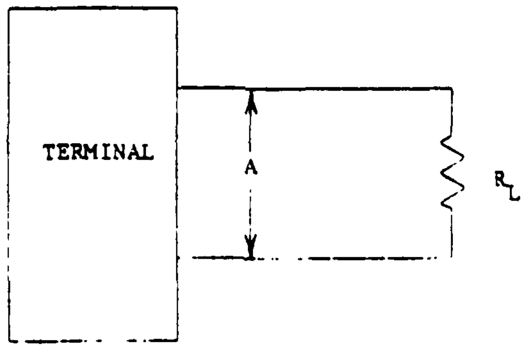


FIGURE 12. Terminal I/O characteristics for transformer coupled and direct coupled stubs.

4.5.1.5.2.2 Cable coupling. All bus-stub junctions shall have continuous shielding which will provide a minimum of 75 percent coverage.

4.5.1.5.2.3 Stub voltage requirements. Every data bus shall be designed such that all stubs at point A of figure 10 shall have a peak-to-peak amplitude, line-to-line within the range of 1.4 and 20.0 V for a transmission by any terminal on the data bus. This shall include the maximum reduction of data bus signal amplitude in the event that one of the terminals has a fault which causes it to reflect a fault impedance of 110 ohms on the data bus. This shall also include the worst case output voltage of the terminals as specified in 4.5.2.1.1.1 and 4.5.2.2.1.1.

4.5.1.5.3 Wiring and cabling for EMC. For purposes of electromagnetic capability (EMC), the wiring and cabling provisions of MIL-E-6051 shall apply.

4.5.2 Terminal characteristics.

4.5.2.1 Terminals with transformer coupled stubs.

4.5.2.1.1 Terminal output characteristics. The following characteristics shall be measured with R_L, as shown on figure 12, equal to 70.0 ohms ± 2.0 percent.

4.5.2.1.1.1 Output levels. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 18.0 to 27.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

4.5.2.1.1.2 Output waveform. The waveform, when measured at point A on figure 12 shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 µs, 1.0 ± .025 µs, 1.5 ± .025 µs, and 2.0 ± .025 µs). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 900.0 millivolts (mV) peak, line-to-line, as measured at point A, figure 12.

4.5.2.1.1.3 Output noise. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed a value of 14.0 mV, RMS, line-to-line, as measured at point A, figure 12.

4.5.2.1.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A of figure 12 shall be no greater than ± 250.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are 8000_{16}, $7FFF_{16}$, 0000_{16}, $FFFF_{16}$, 5555_{16}, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.1.1.1 and 4.5.2.1.1.2.

4.5.2.1.2 Terminal input characteristics The following characteristics shall be measured independently.

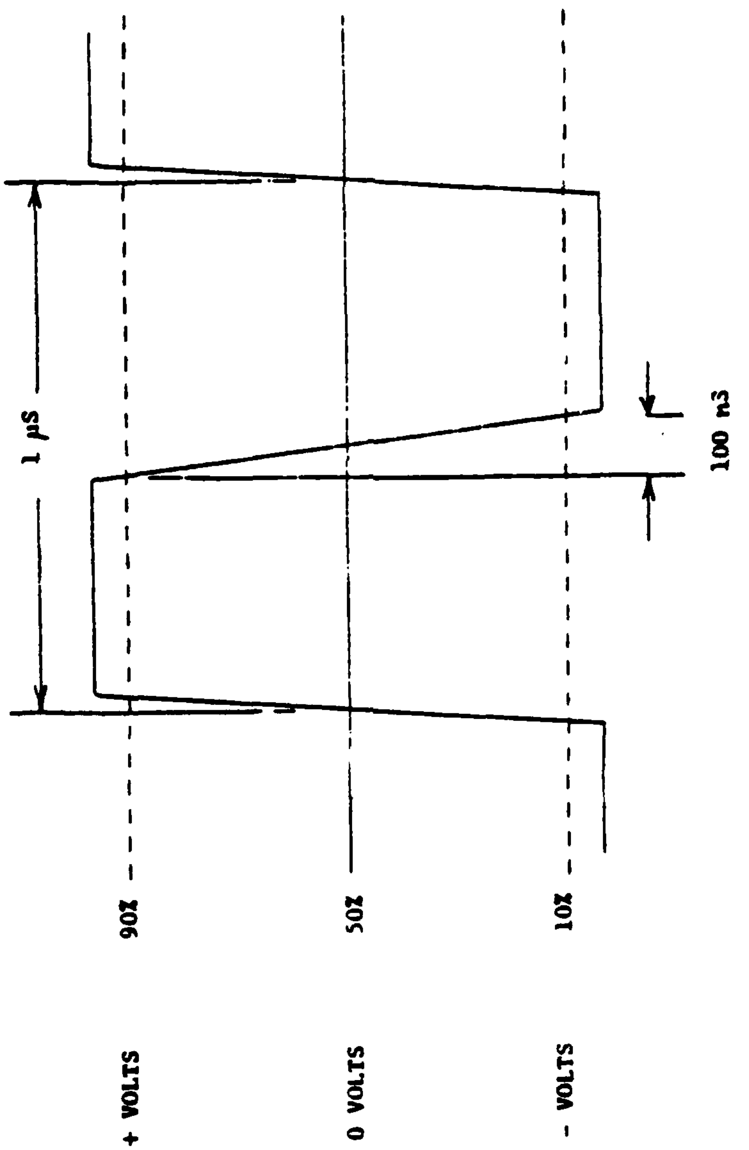


FIGURE 13. Output waveform.

4.5.2.1.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of ± 150 ns, (i.e., 2.0 ± .15 µs, 1.5 ± .15 µs, 1.0 ± .15 µs, .5 ± .15 µs). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of .86 to 14.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .20 V. The voltages are measured at point A on figure 9.

4.5.2.1.2.2 Common mode rejections. Any signals from direct current (DC) to 2.0 MHz, with amplitudes equal to or less than ± 10.0 V peak, line-to-ground, measured at point A on figure 9, shall not degrade the performance of the receiver.

4.5.2.1.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 1000.0 ohms within the frequency range of 75.0 khz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 9.

4.5.2.1.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in 10^7, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 140 mV. A word error shall include any fault which causes the message error bit to be set in the terminal's status word, or one which causes a terminal to not respond to a valid command. The word error rate shall be measured with a 2.1 V peak-to-peak, line-to-line, input to the terminal as measured at point A on figure 9. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

4.5.2.2 Terminals with direct coupled stubs.

4.5.2.2.1 Terminal output characteristics. The following characteristics shall be measured with R_L, as shown on figure 12, equal to 35.0 ohms ± 2.0 percent.

4.5.2.2.1.1 Output levels. The terminal output voltage levels shall be measured using the test configuration shown on figure 12. The terminal output voltage shall be within the range of 6.0 to 9.0 V, peak-to-peak, line-to-line, when measured at point A on figure 12.

Table II. Criteria for acceptance or rejection of a terminal for the noise rejection test

TOTAL WORDS RECEIVED BY THE TERMINAL
(in multiples of 10^7)

No. of Errors	Reject (Equal or less)	Accept (Equal or more)
0	N/A	4.40
1	N/A	5.21
2	N/A	6.02
3	N/A	6.83
4	N/A	7.64
5	N/A	8.45
6	.45	9.27
7	1.26	10.08
8	2.07	10.89
9	2.88	11.70
10	3.69	12.51
11	4.50	13.32
12	5.31	14.13
13	6.12	14.94
14	6.93	15.75
15	7.74	16.56
16	8.55	17.37
17	9.37	18.19
18	10.18	19.00
19	10.99	19.81
20	11.80	20.62
21	12.61	21.43
22	13.42	22.24
23	14.23	23.05
24	15.04	23.86
25	15.85	24.67
26	16.66	25.48
27	17.47	26.29
28	18.29	27.11
29	19.10	27.92
30	19.90	28.73
31	20.72	29.54
32	21.53	30.35
33	22.34	31.16
34	23.15	31.97
35	23.96	32.78
36	24.77	33.00
37	25.58	33.00
38	26.39	33.00
39	27.21	33.00
40	28.02	33.00
41	33.00	N/A

4.5.2.2.1.2 Output waveform. The waveform, when measured at point A on figure 12, shall have zero crossing deviations which are equal to, or less than, 25.0 ns from the ideal crossing point, measured with respect to the previous zero crossing (i.e., .5 ± .025 µs, 1.0 ± .025 µs, 1.5 ± .025 µs and 2.0 ± .025 µs). The rise and fall time of this waveform shall be from 100.0 to 300.0 ns when measured from levels of 10 to 90 percent of full waveform peak-to-peak, line-to-line, voltage as shown on figure 13. Any distortion of the waveform including overshoot and ringing shall not exceed ± 300.0 mV peak, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.3 Output noise. Any noise transmitted when the terminal is receiving or has power removed, shall not exceed a value of 5.0 mV, RMS, line-to-line, as measured at point A on figure 12.

4.5.2.2.1.4 Output symmetry. From the time beginning 2.5 us after the mid-bit crossing of the parity bit of the last word transmitted by a terminal, the maximum voltage at point A on figure 12, shall be no greater than ± 90.0 mV peak, line-to-line. This shall be tested with the terminal transmitting the maximum number of words it is designed to transmit, up to 33. This test shall be run six times with each word in a contiguous block of words having the same bit pattern. The six word contents that shall be used are 8000_{16}, $7FFF_{16}$, 0000_{16}, $FFFF_{16}$, 5555_{16}, and $AAAA_{16}$. The output of the terminal shall be as specified in 4.5.2.2.1.1 and 4.5.2.2.1.2.

4.5.2.2.2 Terminal input characteristics. The following characteristics shall be measured independently.

4.5.2.2.2.1 Input waveform compatibility. The terminal shall be capable of receiving and operating with the incoming signals specified herein, and shall accept waveform varying from a square wave to a sine wave with a maximum zero crossing deviation from the ideal with respect to the previous zero crossing of plus or minus 150 ns, (i.e., 2.0 ± .15 us 1.5 ± .15 us 1.0 ± .15 us .5 ± .15 us). The terminal shall respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 1.2 to 20.0 V. The terminal shall not respond to an input signal whose peak-to-peak amplitude, line-to-line, is within the range of 0.0 to .28 V. The voltages are measured at point A on figure 10.

4.5.2.2.2.2 Common mode rejections. Any signals from DC to 2.0 MHz, with amplitudes equal to or less than ± 10.0 V peak, line-to-ground, measured at point A on figure 10, shall not degrade the performance of the receiver.

4.5.2.2.2.3 Input impedance. The magnitude of the terminal input impedance, when the RT is not transmitting, or has power removed, shall be a minimum of 2000.0 ohms within the frequency range of 75.0 kHz to 1.0 MHz. This impedance is that measured line-to-line at point A on figure 10.

4.5.2.2.2.4 Noise rejection. The terminal shall exhibit a maximum word error rate of one part in 10^7, on all words received by the terminal, after validation checks as specified in 4.4, when operating in the presence of additive white Gaussian noise distributed over a bandwidth of 1.0 kHz to 4.0 MHz at an RMS amplitude of 200 mV. A word error shall include any fault which causes the message error bit to be set in the terminal's status word, or one which causes a terminal to not respond to a valid command. The word error rate shall be measured with a 3.0 V peak-to-peak, line-to-line, input to the

terminal as measured at point A on figure 10. The noise tests shall be run continuously until, for a particular number of failures, the number of words received by the terminal, including both command and data words, exceeds the required number for acceptance of the terminal, or is less than the required number for rejection of the terminal, as specified in table II. All data words used in the tests shall contain random bit patterns. These bit patterns shall be unique for each data word in a message, and shall change randomly from message to message.

4.6 Redundant data bus requirements. If redundant data buses are used, the requirements as specified in the following shall apply to those data buses.

4.6.1 Electrical isolation. All terminals shall have a minimum of 45 dB isolation between data buses. Isolation here means the ratio in dB between the output voltage on the active data bus and the output voltage on the inactive data bus. This shall be measured using the test configuration specified in 4.5.2.1.1 or 4.5.2.2.1 for each data bus. Each data bus shall be alternately activated with all measurements being taken at point A on figure 12 for each data bus.

4.6.2 Single event failures. All data buses shall be routed to minimize the possibility that a single event failure to a data bus shall cause the loss of more than that particular data bus.

4.6.3 Dual standby redundant data bus. If a dual redundant data bus is used, then it shall be a dual standby redundant data bus as specified in the following paragraphs.

4.6.3.1 Data bus activity. Only one data bus can be active at any given time except as specified in 4.6.3.2.

4.6.3.2 Reset data bus transmitter. If while operating on a command, a terminal receives another valid command, from either data bus, it shall reset and respond to the new command on the data bus on which the new command is received. The terminal shall respond to the new command as specified in 4.3.3.8.

5. DETAIL REQUIREMENTS (Not Applicable)

Custodians:
Army - EL
Navy - AS
Air Force - 11

Preparing Activity:
Air Force - 11

Project MISC-0003

APPENDIX

10. General. The following paragraphs in this appendix are presented in order to discuss certain aspects of the standard in a general sense. They are intended to provide a user of the standard more insight into the aspects discussed.

10.1 Redundancy. It is intended that this standard be used to support rather than to supplant the system design process. However, it has been found, through application experience in various aircraft, that the use of a dual standby redundancy technique is very desirable for use in integrating mission avionics. For this reason, this redundancy scheme is defined in 4.6 of this standard. None the less, the system designer should utilize this standard as the needs of a particular application dictate. The use of redundancy, the degree to which it is implemented, and the form which it takes must be determined on an individual application basis. Figures 10.1 and 10.2 illustrate some possible approaches to dual redundancy. These illustrations are not intended to be inclusive, but rather representative. It should be noted that analogous approaches exist for the triple and quad redundant cases.

10.2 Bus controller. The bus controller is a key part of the data bus system. The functions of the bus controller, in addition to the issuance of commands, must include the constant monitoring of the data bus and the traffic on the bus. It is envisioned that most of the routine minute details of bus monitoring (e.g., parity checking, terminal non-response time-out, etc.) will be embodied in hardware, while the algorithms for bus control and decision making will reside in software. It is also envisioned that, in general, the bus controller will be a general purpose airborne computer with a special input/output (I/O) to interface with the data bus. It is of extreme importance in bus controller design that the bus controller be readily able to accommodate terminals of differing protocol's and status word bits used. Equipment designed to MIL-STD-1553A will be in use for a considerable period of time; thus, bus controllers must be capable of adjusting to their differing needs. It is also important to remember that the bus controller will be the focal point for modification and growth within the multiplex system, and thus the software must be written in such a manner as to permit modification with relative ease.

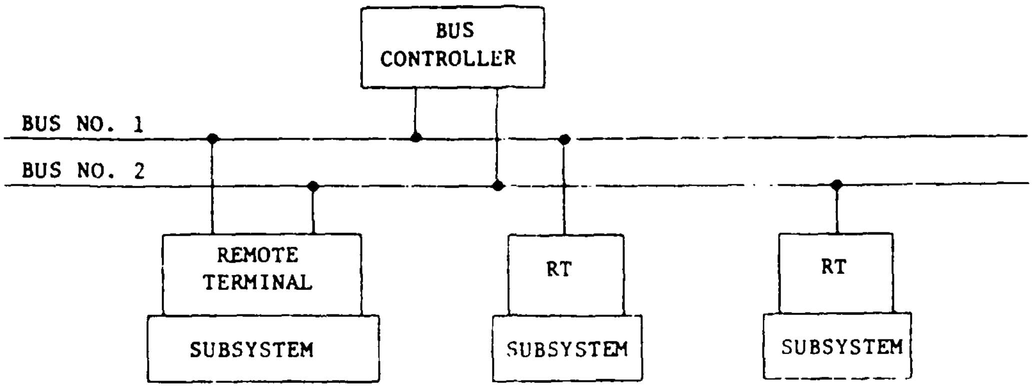


FIGURE 10.1. Illustration of possible redundancy.

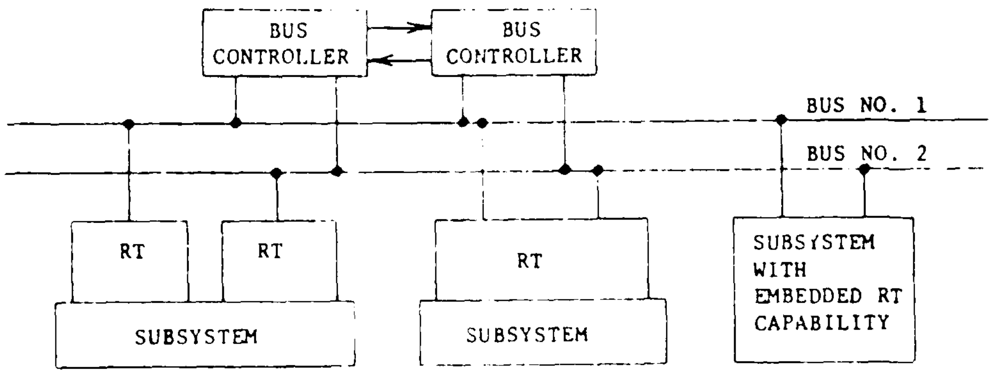


Figure 10.2

NOTE: RT - Remote Terminal

FIGURE 10.2. Illustration of possible redundancy.

10.3 <u>Multiplex selection criteria</u>. The selection of candidate signals for multiplexing is a function of the particular application involved, and criteria will in general vary from system to system. Obviously, those signals which have bandwidths of 400 Hz or less are prime candidates for inclusion on the bus. It is also obvious that video, audio, and high speed parallel digital signals should be excluded. The area of questionable application is usually between 400 Hz and 3KHz bandwidth. The transfer of these signals on the data bus will depend heavily upon the loading of the bus in a particular application. The decision must be based on projected future bus needs as well as the current loading. Another class of signals which in general are not suitable for multiplexing are those which can be typified by a low rate (over a mission) but possessing a high priority or urgency. Examples of such signals might be a nuclear event detector output or a missile launch alarm from a warning receiver. Such signals are usually better left hardwired, but they may be accommodated by the multiplex system if a direct connection to the bus controller's interrupt hardware is used to trigger a software action in response to the signal.

10.4 <u>High reliability requirements</u>. The use of simple parity for error detection within the multiplex bus system was dictated by a compromise between the need for reliable data transmission, system overhead, and remote terminal simplicity. Theoretical and empirical evidence indicates that an undetected bit error rate of $10^{[illegible]}$ can be expected from a practical multiplex system built to this standard. [illegible] a particular signal requires a bit error rate which is better than that provided by the parity checking, then it is incumbent upon the system designer to provide the reliability within the constraints of the standard or to not include this signal within the multiplex bus system. A possible approach in this case would be to have the signal source and sink provide appropriate error detection and correction encoding/decoding and employ extra data words to transfer the information. Another approach would be to partition the message, transmit a portion at a time, and then verify (by interrogation) the proper transfer of each segment.

10.5 <u>Stubbing</u>. Stubbing is the method wherein a separate line is connected between the primary data bus line and a terminal. The direct connection of a stub line causes a mismatch which appears on the waveforms. This mismatch can be reduced by filtering at the receiver and by using bi-phase modulation. Stubs are often employed not only as a convenience in bus layout but as a means of coupling a unit to the line in such a manner that a fault on the stub or terminal will not greatly affect the transmission line operation. In this case, a network is employed in the stub line to provide isolation from the fault. These networks are also used for stubs that are of such length that the mismatch and reflection degrades bus operation. The preferred method of stubbing is to use transformer coupled stubs, as defined in 4.5.1.5.1. This method provides the benefits of DC isolation, increased common mode protection, a doubling of effective stub impedance, and fault isolation for the entire stub and terminal. Direct coupled stubs, as defined in 4.5.1.5.2 of this standard, should be avoided if at all possible. Direct coupled stubs provide no DC isolation or common mode rejection for the terminal external to its subsystem. Further, any shorting fault between the subsystems internal isolation resistors (usually on a circuit board) and the main bus junction will cause failure of that entire bus. It can be expected that when the direct coupled stub length exceeds 1.6 feet, that it will begin to distort the main bus waveforms. Note that this length includes the cable runs internal to a given subsystem.

10.6 Use of broadcast option. The use of a broadcast message as defined in 4.3.3.6.7 of this standard represents a significant departure from the basic philosophy of this standard in that it is a message format which does not provide positive closed-loop control of bus traffic. The system designer is strongly encouraged to solve any design problems through the use of the three basic message formats without resorting to use of the broadcast. If system designers do choose to use the broadcast command, they should carefully consider the potential effects of a missed broadcast message, and the subsequent implications for fault or error recovery design in the remote terminals and bus controllers.

CHAPTER 11

PARAMETER FORMATS

11.0 PARAMETER FORMATS

(To be added on a later revision)

CHAPTER 12
INDEX

12.0 INDEX

(To be added on a later revision)

APPENDIX A

DATA BASE ANALYSIS TOOLS

TABLE OF CONTENTS

APPENDIX A

TOOLS NEEDED FOR DATA BASE ANALYSIS

1.0 INTRODUCTION

Two computer programs to assist the designer have been developed under Air Force Avionics Laboratory (AFAL) contracts. Conceptually, these programs only accomplish part of the job that has always been required of the designer. However, through the use of software simulation, the system designer can be relieved of the laborious tasks and allowed to establish the design. The two programs are the Multiplex Systems Simulator (MUXSIM) and the Standard Interface Applicability Analysis Program (SIAAP). MUXSIM was developed under AFAL contract F33615-76-C-1172 by the Harris Corporation of Melbourne, Florida. SIAAP was developed under AFAL contract F33615-73-C-1222 by SCI Systems, Inc. of Huntsville, Alabama. Operational evaluation of these software systems yielded positive results, proving these extensive design tools are necessary to adequately investigate architectural alternatives.

2.0 WHAT IS MUXSIM?

MUXSIM is a software package that performs computer-aided design and design verification of digital information transfer systems. It is primarily an aid for an organized approach to specify and design multiplex systems for diverse applications. MUXSIM is a useful tool for determining answers to complex digital multiplex system implementation problems such as the following:

a. Command and control techniques for data transfer
b. Data bus requirements
c. System interface hardware requirements
d. Data processing and manipulation requirements

MUXSIM is a software system that supports the data base, defines all the detailed signal transfer and interconnect requirements needed to specify the data network, provides techniques to extract the necessary engineering information from the data base, and weighs selected data network approaches against the requirements.

MUXSIM can be best defined with a brief description of the typical outputs generated while analyzing a problem:

a. Corrected equipment list, signal list, and signal summaries

b. Remote terminal equipment interconnect requirements

c. Word and message structure and processing requirements to accomplish the data transfer

d. Average bus utilization reports

e. Fixed format bus schedule

f. Time delay and message queuing statistics

Coaching messages, error messages and editing information are made available via prompting to instruct and advise the user.

MUXSIM consists of four major subsystems: utility, static, dynamic, and executive. These subsystems are divided into a total of 26 interrelated programs accessed only through the executive program. This program provides interactive prompting so that users can easily acquaint themselves with the system. Currently, this program operates on a mainframe machine (DEC System 10) with an interactive terminal. The program also operates on a minicomputer (Datacraft) in a batch mode. Operation in this mode requires the analyst to act as the manual executive.

The utility subsystem is essentially the data base management system for MUXSIM. It provides the tools to support the generation of the data base and provides the accounting routines to check the data base for completeness. These checks consist of the equipment complement versus the signal list information and all input signals versus output signals for completeness. The programs flag all equipment and signal deficiencies for designer review.

The utility subsystem contains programs to--

a. Translate human-readable signal list to machine usable signal list

b. Reconstruct human-readable signal list from machine-usable signal list

c. Test signal list for inconsistencies and errors; test signal list for inconsistencies against equipment complement and report inconsistencies in human readable format

d. Provide interactive edit of signal list

e. Modify signal list of conventional sensors to new sensors postulated by ·gital avionics integration schemes

The utility package enables the user to create, edit, update, and refine the signal list that defines the problem or data transfer requirements.

The static subsystem consists of the remote terminal (RT) assignment, word map, fixed-format scheduling and fixed format bus loading computation. Eight models (representing eight distinct data transfer system message formats) are implemented in this subsystem. It is designated "static" because the subsystem addresses the stationary aspects of bus traffic analysis (e.g., RT interconnections, format schedules for periodic data transmission, computing average bus loading and utilization).

The static subsystem contains programs to--

a. Provide signal accounting by signal type and geographic location for each remote terminal

b. Provide totals and subtotals necessary to accomplish RT assignment

c. Convert line-replaceable-unit (LRU) to RT assignment to signal to RT assignment. The LRU to RT assignments are made by using a brief dictionary

d. Provide interactive editing of signal to RT assignment to "fine-tune" the assignment implemented through the dictionary

e. Provide documentation (I/O control document or specification quality) of detailed RT to signal interface, specifically showing all interfaces at each remote terminal

f. Provide grouping of signals by each data word on the data bus (word mapping) as a function of the control mechanism selected

g. Provide grouping of words (signals) by each data message on the bus (message mapping) according to the control mechanism selected

h. Provide documentation of the signal-to-message assignment

i. Assign periodic data messages to minor frame sequences by their update requirements and compute bus utilization for each minor frame

j. Compute average loading and utilization of the bus

The eight distinct message format models presently implemented are:

a. Terminal-terminal transfer

b. Terminal-central-terminal transfer, with the central doing bit manipulation

c. Terminal-terminal transfer of word data, with discrete data handled via terminal-central-terminal and the central doing bit manipulation

d. Hybrid of both terminal-terminal and terminal-central-terminal transfer, with the choice dependent on the lower bus loading for each data update rate

e. Terminal-terminal transfer employing some smart terminals that are capable of selecting their data from among the transmissions (broadcast reception)

f. Terminal-central-terminal employing some smart terminals using broadcast reception

g. Hybrid combination of the two prior schemes, using lower bus loading as a selection criterion for each update rate.

h. Terminal-central-terminal, with central being restricted to word manipulation, thereby requiring the sending terminals to pack data compatible with the receiving terminal.

Selection of these eight message format approaches was influenced by MIL-STD-1553 message format considerations.

The dynamic subsystem consists of two discrete-event models and is termed "dynamic" because stochastic events such as multiplex system component failures, bus noise, and time-variable data transfer requirements are considered in the analysis. Because it uses a complete discrete-event and continuous simulation package (GASP IV) as a component, the system is general purpose.

The discrete-event models simulate in time such events as:

a. Demand message arrival
b. Demand message queue
c. Fixed-schedule message queue
d. Minor frame clock
e. Start next message transmission
f. Terminal transmit request
g. Data bus noise
h. Terminal response
i. Permanent terminal failure
j. Intermittent terminal failure
k. Recovery from intermittent terminal failure
l. Terminal response time out circuitry (watchdog)

3.0 WHAT IS SIAAP?

The Standard Interface Applicability Analysis Program (SIAAP) is a computer-based analysis program used to match standard electrical interface circuitry with actual electrical interfaces exhibited by equipment. The program uses basic electrical circuit analysis to determine a match or mismatch of the actual signals against the defined standard interface circuitry. These consist of:

a. Type of interface
b. Loading effects
c. Maximum voltage effects
d. Accuracy
e. Resolution
f. Maximum current effects

Using this electrical analysis technique, specific hardware can be judged in terms of interface compatibility.

Because of SIAAP's basic emphasis, it can be programmed on a variety of computing devices. SIAAP has been used most effectively on two types of machines, a very large mainframe machine (CDC 6600) and a handheld calculator (HP 25). Obviously, in each of these modes it provides different useful features. In the calculator mode, it provides a simple method of checking a few electrical characteristics of a signal to determine which standard interface it matches or what modifications are necessary to the standard interface to provide an electrical match. However, SIAAP is primarily used on large signal integration jobs requiring the mainframe machine because bookkeeping is essential to understanding the problem.

An understanding of SIAAP can be expressed from a user viewpoint by listing the output reports it generates:

a. Number of matched signals by location

b. Numerical summary of matched signals

c. Number of matched signals by subsystem interface module type (standard interface circuitry)

d. Number of subsystem interface modules (SSIMs)

e. Number of SSIM's required, by SSIM type

f. Numerical summary of SSIM requirements

g. Achieved utilization factor by location

h. Achieved utilization factor by SSIM

i. Summary of achieved utilization factor

j. Listing of mismatched signal characteristics and locations

k. Numerical summary of mismatched signals by SSIM type

l. Signal listing with location and SSIM type for matched signal and locations for mismatched signals

m. Signal listing for matched signals by location

With these output routines, the systems engineer has the capability to optimize the number and types of interfaces required for the integration. After optimization is complete, the output of these reports can be used to identify physical separation requirements to satisfy redundancy and survivability standards. From the data for each interface a hardware specification can be prepared describing remote terminal interface needs, established wire routing requirements, and integration areas specified within the vehicle. At this point SIAAP has aided the systems engineer in the early phases of program definition, design, and specification. SIAAP can then continue its bookkeeping assistance through the manufacturing phase by evaluating changes as the design evolves to implementation.

APPENDIX B

DATA BUS USE ANALYSIS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

APPENDIX B

DATA BUS USE ANALYSIS

1.0 INTRODUCTION

Use of data bus resources requires careful management. Since multiplex system overhead (mode control and BIT), data bus message format overhead (command/status words), and avionic system data are involved, analysis of the bus resources must be started early in the definition stage of a system and maintained throughout the development and acquisition period and well into the postdelivery time period. Some basic analyses must be accomplished to obtain sufficient visibility of resource use:

a. Bus loading--actual transmissions as a percentage of the maximum possible (allowable) transmissions. Both data and overhead are involved in this calculation.

b. Efficiency--the ratio of data bits transmitted to the total number of bits transmitted.

c. Latency--the delay time that occurs from the initialization of an event to the detection and transmission of the event data to its final destination.

To illustrate the type of analysis that is initially required, the following examples of three multiplex control mechanisms are provided:

a. Stationary master
b. Nonstationary master (polling)
c. Nonstationary master (round robin)

2.0 DATA BUS ANALYSIS

The control bits, words, and messages associated with information transfer system management reduce effective data transfer capability. Since the data bus transmission rate is 1M bits/sec and the word length is 20 bits/word, the effective word transfer rate is 50,000 words/sec. This word transfer rate is reduced by the following overhead functions:

a. Variable message length

b. 16 bits of data per 20-bit data word

c. Message type

d. Command and status word

e. Intermessage gap

f. Response time gap

g. Control messages
   1. Asynchronous transactions
   2. System status checks
   3. Error handling and recovery

Since message lengths are variable (1 to 32 words), the overhead associated with message lengths is an important factor in computing data bus efficiency. Therefore, table B-1 is provided for computing bus efficiency as a function of message length and type of message. This table considers word length, 16 bits of data per 20 bits in a data word, message type, command and status words, intermessage gap, and response time. By examining the table and summing the times for each message in a system, the total message traffic can be calculated in terms of time. The ratio of data bits divided by the number from the table gives data transfer efficiency. The table value also gives the total bus utilization if the number is considered to be in terms of microseconds used per second.

The following equations were developed for the 1553B message formats:

P1 Master Bus Control to Remote Terminal
Remote Terminal to Master Bus Controller

40	+	20	+	20n	= usec
1 command, 1 status		1 response time, 1 intermessage gap		No. of data words n = 1-32	Message time

P2 Remote Terminal to Remote Terminal

80	+	30	+	20n	= usec
2 commands 2 status		2 response times, 1 intermessage gap		No. of data words	Message time

P3 Master Bus Controller to Remote Terminal (Broadcast)

20	+	10	+	20n	= usec
1 command		1 intermessage gap		No. of data words	Message time

P4 Remote Terminal to Remote Terminals (Broadcast)

60	+	20	+	20n	= usec
2 commands, 1 status		1 response time, 1 intermessage gap		No. of data words	Message time

P5 Mode Command Without Data Word

40	+	20	= 60 usec
1 command, 1 status		1 response time, 1 intermessage gap	Message time

P6 Mode Command With Data Word

40	+	20	+	20	= 80 usec
1 command, 1 status		1 response time, 1 intermessage gap		1 data word	Message time

P7 Mode Command (Broadcast) Without Data Word

20	+	10	= 30 usec
1 broadcast command		1 intermessage gap	Message time

P8 Mode Command (Broadcast) with Data Word

20	+	10	+	20	= 50 usec
1 broadcast command		1 intermessage gap		1 data word	Message time

Notes:

1. Response time is assumed to average 10 usec for this analysis. MIL-STD-1553B specifies a response time of 4 to 12 usec.

2. Intermessage gap is assumed to average 10 usec for this analysis. MIL-STD-1553B specifies a minimum intermessage gap of 4 usec.

Table B-1. Type of Message

No. of data words	P1* MBC to RT, RT to MBC (µs)	P2* RT to RT (µs)	P3* MBC to RT(s) (µs)	P4* RT to RT(s) (µs)	P5* Mode command without data word (µs)	P6* Mode command with data word (µs)	P7* Broadcast mode command without data word (µs)	P8* Broadcast mode command with data word (µs)
1	80	130	50	100	60	80	30	50
2	100	150	70	120				
3	120	170	90	140				
4	140	190	110	160				
5	160	210	130	180	**	**	**	**
6	180	230	150	200				
7	200	250	170	220				
8	220	270	190	240				
9	240	290	210	260				
10	260	310	230	280				
11	280	330	250	300				
12	300	350	270	320				
13	320	370	290	340				
14	340	390	310	360				
15	360	410	330	380				
16	380	430	350	400				
17	400	450	370	420				
18	420	470	390	440				
19	440	490	410	460				
20	460	510	430	480				
21	480	530	450	500				
22	500	550	470	520				
23	520	570	490	540				
24	540	590	510	560				
25	560	610	530	580				
26	580	630	550	600				
27	600	650	570	620				
28	620	670	590	640				
29	640	690	610	660				
30	660	710	630	680				
31	680	730	650	700				
32	700	750	670	720	60	80	30	50

*Equations for message types (P1 through P8) are presented in section 2.0.

**Not a function of data words.

2.1 STATIONARY MASTER ITS ANALYSIS

The stationary master information transfer system (ITS) efficiency, trigger message detection speed, and percentage of bus utilization are shown in figures B-1, B-2, and B-3.

The efficiency is computed using a percentage mix of message types that could be expected in an avionic system. Table B-2 defines the eight message types and the percentage of bus traffic assumed for each type (usually the actual vehicle multiplex system message mix would be used). In an attempt to cover the range of message mixes, seven different percentages mixes were selected for analysis, as shown in table B-2. Figure B-1 shows the best and worst case efficiencies associated with these percentage mixes.

The equation for the stationary master efficiency is

$$E = \frac{0.8\,(P_1 + P_2 + P_3 + P_4)\,N}{(N + 3)\,P_1 + (N + 5.5)\,P_2 + (N + 1.5)\,P_3 + (N + 4)\,P_4 + 3P_5 + 4P_6 + 1.5P_7}$$

where

N is the average number of data words per message.

The average response time for trigger messages in the stationary master is shown in figure B-2 using the nominal mix 1. The percentage of bus utilization as a function of load is shown in figure B-3. These data were developed to indicate the capacity of the stationary master using the nominal mix.

2.2 NONSTATIONARY MASTER (POLLING) ITS ANALYSIS

The same parameters that effect efficiency of the stationary master ITS also effect the nonstationary master polling ITS, with the addition of the polling of potential bus controllers, the polling of remote terminal or the polling of terminals for trigger messages, and the bus control transfer time. The additional overhead apparent in the nonstationary master is a function of the number of potential bus controllers and the number of devices that need to be polled for trigger messages. The additional overhead is equal to 10 words plus 5 data words times the number of potential bus controllers (5C+10 where C is the number of potential bus controllers) for the polling of potential bus controllers. Evaluation of the polled data by the processor and the time required for the new processor to assume control after command was assigned to be 125 usec. An additional overhead associated with the nonstationary master is the polling of devices to determine if trigger messages need to be serviced. This overhead is 10 data words plus 5 data words times the number of devices polled plus the time required to evaluate the polled data and the time the devices need to assume control of the bus.

The amount of device polling for trigger messages can go to two extremes: (1) polling occurs after each bus controller message sequence has completed and (2) there is no polling for trigger messages. In the second case, trigger messages are taken care of during normal processing. An alternative approach to the processing of trigger messages is to poll the devices at a fixed time interval (i.e., 10 ms or once per minor cycle).

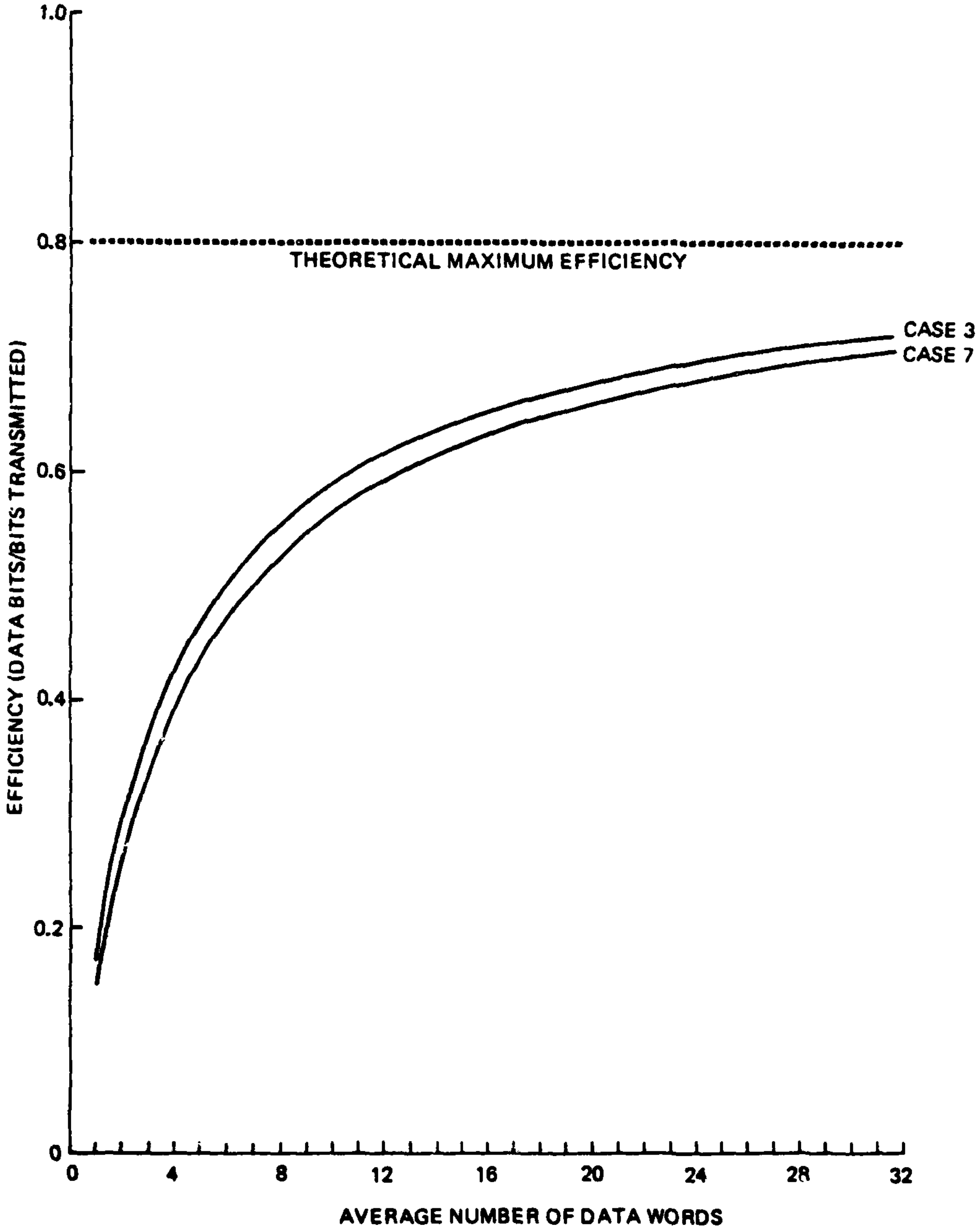


Figure B-1. Stationary Master Efficiency

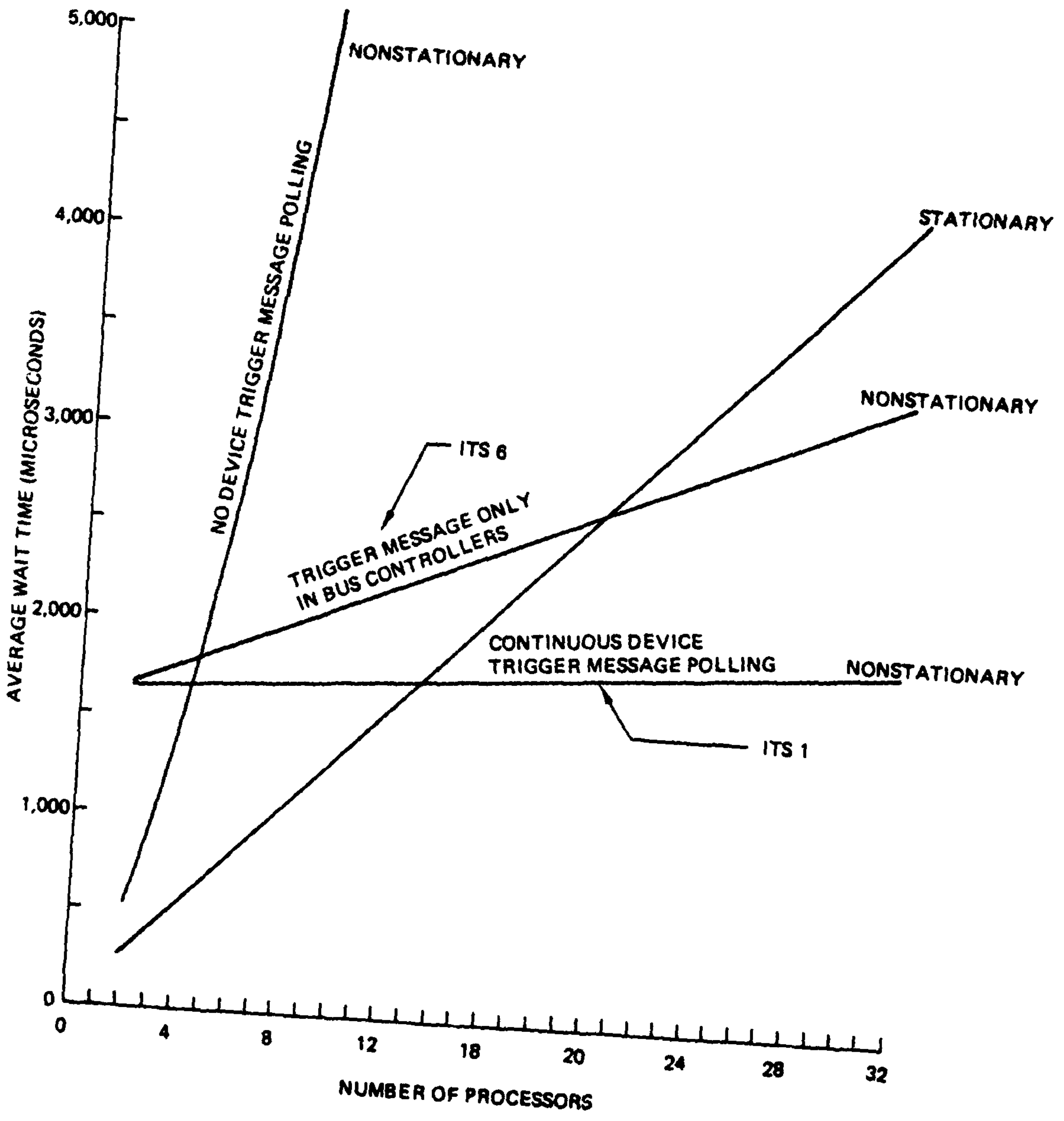
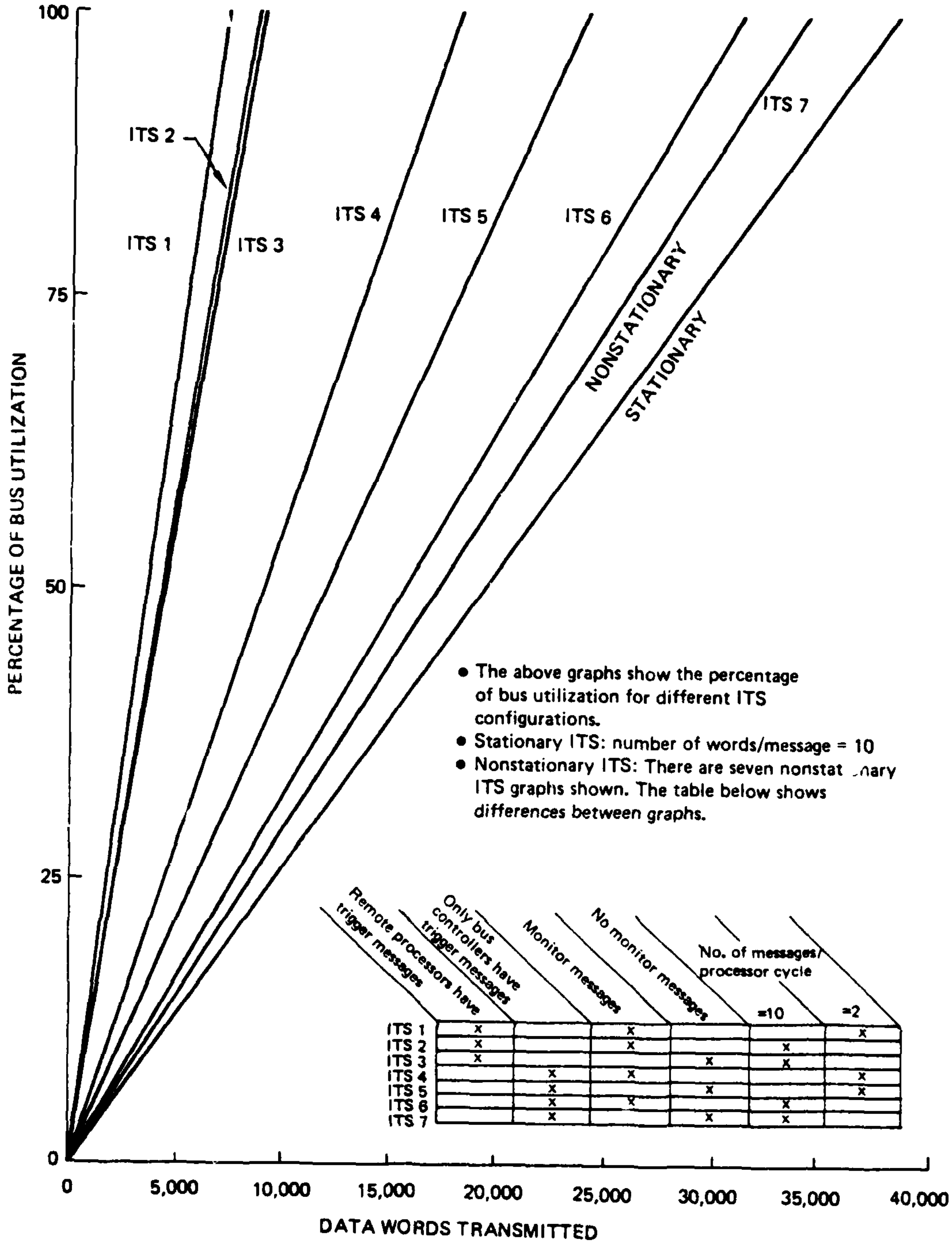


Figure B-2. Average Wait Time for Trigger Messages



	Remote processors have trigger messages	Only bus controllers have trigger messages	Monitor messages	No monitor messages	No. of messages/processor cycle =10	=2
ITS 1	x		x			x
ITS 2	x		x		x	
ITS 3	x			x	x	
ITS 4		x	x			x
ITS 5		x		x		x
ITS 6		x	x		x	
ITS 7		x		x	x	

Figure B-3. Percentage of Bus Utilization

Table B-2. Typical Bus Message Type Mix

	Message type	Percentage of mix						
		(1%)	(2%)	(3%)	(4%)	(5%)	(6%)	(7%)
P_1	MBC to RT RT to MBC	50	35	65	60	55	45	40
P_2	RT to RT	30	40	20	25	20	25	45
P_3	MBC to RT(s) (broadcast)	10	15	5	8	8	15	6
P_4	RT to RT(s) (broadcast)	5	4	3	2	8	7	2
P_5	Mode command without data word	2	1	2	3	3	2	2
P_6	Mode command with data word	1	2	2	0	2	2	2
P_7	Mode command without data word (broadcast)	1	2	2	0	2	2	1
P_8	Mode command with data word (broadcast)	1	1	1	2	2	2	2

MBC Master bus controller

RT Remote terminal

The equation to calculate efficiency for a nonstationary master is

$$E = \frac{.8(P_1+P_2+P_3+P_4)\ N}{(N+3)P_1+(N+5.5)P_2+(N+1.5)P_3+(N+4)P_4+3P_5+4P_6+1.5P_7+2.5P_8+5C+10+250/20+K(5R+10+125/20)}$$

where

R is the number of devices with potential trigger messages.
K represents how often polling the remotes for trigger messages occurs.

$$K = \frac{W + 20 \text{ msec/word } (5C + 10) + 125}{T}$$

where

W is the time required for a master to do its processing. C is the number of potential bus controllers. T is the cycle time (interval) associated with polling devices for trigger messages. The case where polling occurs after every bus controller usage, K = 1; when no device polling is used, K = 0.

Figure B-4 shows how the efficiency is affected when K = 0 or K = 1. Figure B-5 shows how the efficiency is affected when there is a time delay before the device polling sequence occurs. T is the delay time in milliseconds.

The nonstationary master bus control allocation procedure requires a trade between rapid detection and transmission of trigger messages versus the effect on bus efficiency and percentage of bus utilization. Obviously, the best bus efficiency occurs when there is no polling of devices for trigger messages. In this case, the trigger message is processed with no attempt to speed up the transmission; however, this method produces slow responses as the number of bus users increases as shown in figure B-2. An alternative approach is to poll all the devices after each bus controller has completed its transmission sequence. This method improves trigger message response time, as shown in figure B-2, but greatly reduces the efficiency of the bus (see fig. B-3). Another approach to these two methods is to poll devices in a scheduled fashion (once every time interval or 10 ms). These results are shown in figure B-5. This method also encounters problems regarding percentage bus utilization, as shown in figure B-3, ITS 1. Therefore, another approach may be required to improve nonstationary master ITS performance. If the trigger messages were restricted to only potential bus controllers, the number of devices to be polled would be reduced and a trigger message holder would become the next controller. Another potential bus efficiency improvement could be the removal of all overhead traffic to the monitor. Additional benefits can be gained by increasing the average number of messages transmitted by a processor during bus control from 2 to 10 data words per message; however, this is strictly application dependent. Effects on the ITS's percentage bus utilization for each alternative are shown in figure B-3.

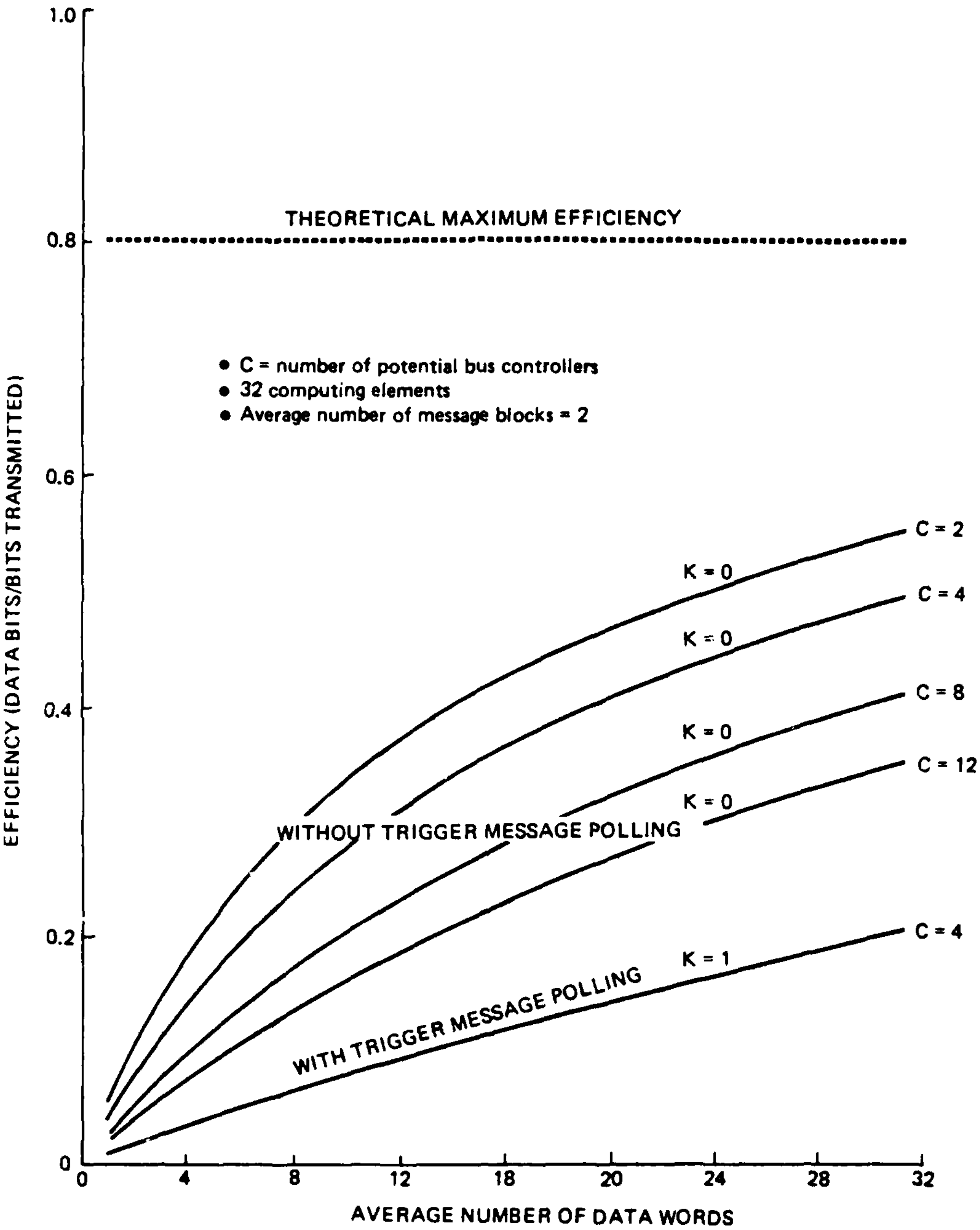


Figure B-4. Trigger Message Polling Effect for Nonstationary Master (Polling)

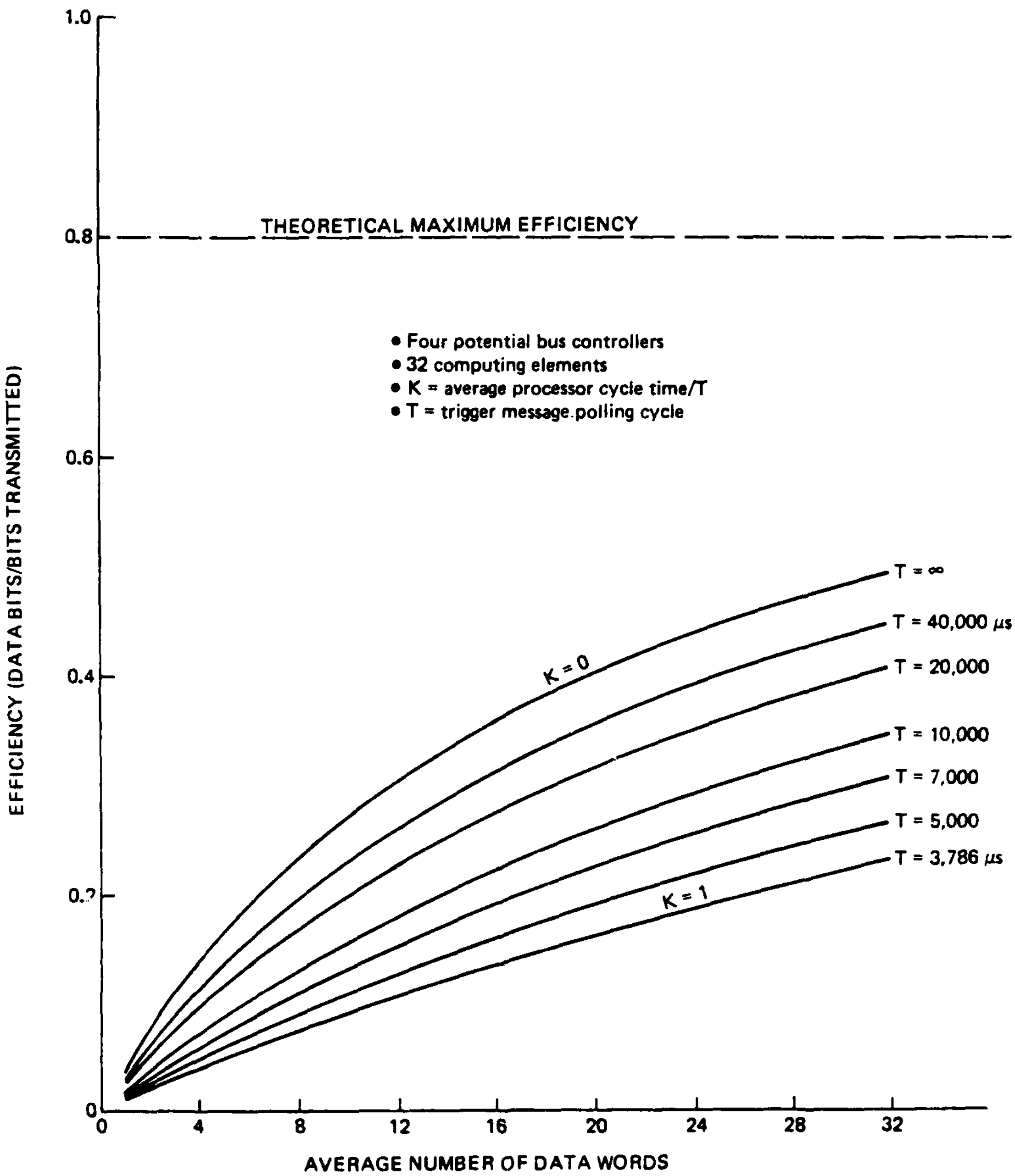


Figure B-5. Nonstationary Master (Polling)

The equation for bus utilization is

$$\% \text{ bus utilization} = \frac{\text{Total traffic on bus}}{\text{Maximum allowable traffic on bus}} 100$$

$$\text{Total Traffic on Bus} = A + \frac{AM20}{T} \; (4R\;(20)) \;\; \frac{X}{N}$$

where

$$A = P_1(N+3) + P_2(N+5) + P_3(N+1.5) + P_4(N+4) + 3P_5 + 4P_6 + 1.5P_7 + 2.5P_8 + \frac{5C+10}{M}$$

where

C is the number of potential bus controllers

M is the average number of messages transmitted by the bus controller during control

N number of data words in a message

R is the number of devices

T is the cycle time between polling the devices for trigger messages

X number of data words transmitted per second

The equation for the total traffic on the bus varies for each ITS type. When the trigger messages are restricted to potential bus controllers, the factor R is zero. When no messages are sent to the monitor, the term 5C+10 is changed to 4C. Figure B-2 shows considerable improvement in bus utilization when trigger messages are restricted to potential bus controllers. The trigger message response time is also within acceptable limits (shown in ITS 6 trigger response time in fig. B-2).

2.3 NONSTATIONARY MASTER (ROUND ROBIN) ITS ANALYSIS

The same parameters that affected efficiency of the nonstationary master, polling ITS will affect to a limited extent, the nonstationary master round-robin system. The overhead associated with the round robin system is identical to the stationary master with the additional overhead associated with the transfer of bus controller. In this example analysis, no polling of trigger message is considered. If polling for a trigger message is desired, the discussion in the nonstationary master polling scheme is applicable. The polling required to accomplish a bus controller transfer requires 8 data words and some delay time that is assumed to be 200 usec.

The equation for the efficiency of the round-robin ITS is shown below; the variable names are similar to those defined for the stationary master ITS example.

$$E = \frac{0.8(P_1 + P_2 + P_3 + P_4)N}{P_1(N+3)+P_2(N+5.5)+P_3(N+1.5)+P_4(N+4) + 3P_5 + 4P_6 + 1.5P_7 + 2.5P_8 + \frac{200\ \text{usec}}{K}}$$

where N is the average number of data words per message.

$$K = \frac{\text{The number of data words transmitted per processor bus control cycle}}{\text{The number of data words in a standard data transmission mix}}$$

The same standard data transmission mix used previously was assumed for the denominator of the efficiency equation without the f(K) portion.

A severe limitation of the round-robin ITS is its inability to handle asynchronous message transmissions rapidly. The equation below calculates the average wait time for asynchronous messages.

$$W = \frac{A\ (B - 1)}{2} + D$$

Note: This equation applies for two or more processors where

- W average wait time for a trigger message
- A average time required for a processor to complete its processing
- B number of processors capable of controlling the bus
- D time it takes an active bus controller to start transmitting an asynchronous message

3.0 RESULTS OF MULTIPLEX SYSTEM ANALYSIS USING DATA BASE ANALYSIS TOOLS

The analysis discussion covered two approaches to analyzing a multiplex system in its preliminary form. Detailed analysis will be based on the software design tools discussed earlier (appendix A). To illustrate the benefit of this detailed analysis that is required during system design and definition, several Standard Interface Applicability Analysis Program (SIAAP) and Multiplex Systems Simulator (MUXSIM) analyses are discussed.

3.1 SIAAP ANALYSIS TOOL

The results of a SIAAP analysis can best be described by examining the simulation tools' output printouts. Figure B-6 shows the signals that interface with the remote terminals defined for a given zone. These particular signals interface with the passive discrete module. These data can be used for final interconnections within the vehicle. In contrast to the matched interfaces, a set of interfaces that is not compatible with the system interface hardware is defined in figure B-7. This list has been generated as an exception list defining all signals within a zone that will not interface with the remote terminals within that zone. Each of these exceptions must be handled on an individual basis. In some cases, the signal may be routed to other terminals in adjacent zones with this interface or the signal may require a special interface.

The key to a successful integration is the effective use of the integration hardware by interface type and location. This analysis is accomplished by SIAAP, and the results are presented in several basic formats:

a. Module count by module type for a given remote terminal (fig. B-8)
b. Total module count by module type (fig. B-9)
c. Module type by remote terminal (fig. B-10)
d. Utilization factor of module types by terminal (fig. B-11)
e. Utilization factor of a given module by terminals (fig. B-12)
f. Total utilization factor by module type (fig. B-13)

3.2 MUXSIM ANALYSIS TOOL

MUXSIM analysis results can also be described by examining the simulation tools' output printouts. Figure B-14 shows the remote terminal interface for a given set of signals. This analysis assumes that the proper interface hardware is available and that the signals have been mapped to this particular terminal based on geography within the vehicle. Because of this assumption, the SIAAP and MUXSIM analyses are supportive of each other rather than competitive. Analysis of the data bus traffic by MUXSIM is summarized in figure B-15, giving the update rates and the summation of word/sec associated with each update rate. The bus resource utilization represents average bus usage. Details of the message distribution appear in figures B-16, B-17, B-18, B-19, and B-20. The bus schedule printout identifies the message list against the update rates. This can be further described by a human-readable list of each signal within the message list (fig. B-17). This can be further broken down to the message flow summary and word flow summary shown in figures B-19 and B-20.

One detailed analysis that must be accomplished as system design continues deals with the minor cycle bus loading. Rather than the relatively simple

total bus loading, this analysis considers the effect of updating on an individual message and the mapping of these data into minor cycles (frames). The purpose of this analysis is to alert the designer to potentially highly used minor frames that may require some messages be moved to less active minor frames while still maintaining the update rate requirements of the parameters. Note that minor frame loading will not be the same unless all minor frames have the same message list. Figure B-18 shows a typical minor frame loading. This example represents an acceptable condition, and little effort should be expended to balance these loads. The only concern the designer should have is that the minor frame does not exceed a maximum of 80% of the time allowed for the minor frame (good design practice) to allow future messages to occupy the minor frame.

SIMCOM DATE PROCESSED 9/2/75 PAGE 1

BOEING 747 BOAC RA316

REMOTE TERMINAL ZONE 4

SUBSYSTEM INTERFACE MODULE PASSIVE DISCRETE INPUT 2
CHARACTERISTICS:

LOGIC ONE THRESHOLD VOLTAGE	14.0
LOGIC ZERO THRESHOLD VOLTAGE	9.0
INPUT IMPEDANCE	100K
MAXIMUM VOLTAGE	36.0
8 CHANNELS PER MODULE	

SOURCE		DWG. NO.	SIGNAL NAME	SINK		CL	S/WD	S/S		INTERFACE SOURCE			INTERFACE SINK				C	SEQ NO.
DX 25	B	256311	EMER EVAC ON SIG	COMP25		LD	01	001	18	0	0	1					1	1758
M 205	AP	303111	PROBE HT SL SW AP	COMP30		LD	01	005	28	0	0350						1	3973
M 205	TP	303111	PROBE HT SL SW TP	COMP30		LD	01	005	28	0	0350						1	3874
M 205	MP	303111	PROBE HT SL SW MP	COMP30		LD	01	005	28	0	0350						1	3875
B00080		234211	AURAL WARNING	M00470	7	LD	01	5	15	0	5	1	15	5	300	30	1	90
M00197		234311	PASS CALL SOFTWARE	R00960	A3	LD	01	5	15	0	5	1	15	5	50	30	1	317
B00021	5	232811	SELCAL RESET	B00020	5	LD	01	20	18	0	0	1	18	3	600	29	1	950
B00021	6	232811	SELCAL RESET	B00020	20	LD	01	20	15	0	0	1	18	0	600	29	1	951
C00282	Z	233411	ENTERTAIN TAPE	R00520	X1	LD	01	50	18	0	0	1	18	3	600	29	1	1150
B00114	SF	344112	ATTITUDE MODE CON	B00109F	51	LD	01	1	18	0	0	1	18	0	10K	29	1	4175
B00114	SE	344112	STANDBY OR ALIGN	B00109F	50	LD	01	1	18	0	0	1	18	0	10K	29	1	4177
B00114	V	344112	STANDBY	B00109	47	LD	01	1	15	0	0	1	18	0	10K	29	1	4179
B00114	U	344112	MSU POWER ON	B00109	46	LD	01	1	18	0	0	1	18	0	10K	29	1	4192
B00115	V	344122	STANDBY	B00110F	47	LD	01	1	18	0	0	1	15	0	10K	29	1	4291
B00115	SE	344122	STANDBY OR ALIGN	B00110F	50	LD	01	1	15	0	0	1	18	0	10K	29	1	4293
B00115	SF	344122	ATTITUDE MODE CON	B00110F	51	LD	01	1	18	0	0	1	18	0	10K	29	1	4294
B00172	V	344132	STANDBY	B00173F	47	LD	01	1	18	0	0	1	18	0	10K	29	1	4333
B00172	SE	344132	STANDBY OR ALIGN	B00173F	50	LD	01	1	18	0	0	1	15	0	10K	29	1	4335
B00172	SF	344132	ATTITUDE MODE CON	B00173	51	LD	01	1	18	0	0	1	15	0	10K	29	1	4336
S00917	1	344313	ALERT INDICATOR	L01754	1	LD	1	5	18	0	0	1	18	0	600	29	1	4613

Figure B-6. Matched Signal by SSIM and Terminal

SIMCOM 46-1 DATE PROCESSED 10/16/75 PAGE 2

BOEING 747 BOAC RA316

REMOTE TERMINAL ZONE: 9

SUBSYSTEM INTERFACE MODULE: NO MATCH OUTPUT

CHARACTERISTICS:

DOES NOT MATCH WITH ANY AVAILABLE MODULE

SOURCE		DWG. NO.	SIGNAL NAME	SINK		CL	B/WD	S/S	INTERFACE SOURCE	INTERFACE SINK	C	SEQ NO.
B00348C	21	345811	SLEW L-R CONTROL	B00347B	21	LD	1	5	18 C 0 0	18 1 600 20	1	5032
B00348C	22	345811	FCN 1-2-3	B00347B	22	LD	1	5	18 0 1 1	18 1 600 20	1	5033
B00348C	24	345811	DECADE 4 CTRL	B00347B	24	LD	1	5	18 0 1 1	18 1 600 20	1	5034
B00348C	27	345811	CHANNEL CONTROL	B00347B	27	LD	1	5	18 0 1 1	18 1 600 20	1	5036
B00348C	29	345811	AFC CONTROL	B00347B	28	LD	1	5	18 C 1 1	18 1 600 20	1	5037
COMP24		243211	PC3032 CONTROL	PC3032	4	LD	01	001		3.5 2.5 166 8	1	783
COMP24		243211	PC3033 CONTROL	PC3033	4	LD	01	001		3.5 2.5 166 8	1	792
COMP24		243211	PC3030 CONTROL	PC3030	4	LD	01	001		3.5 2.5 166 8	1	865
M 1525	3	283121	NO2 18 JETT PU SW	R 42	X1	LD	01	005	0 0 C 1	C.2 10K 600CC	1	2935
M 1525	2	283121	NC3 18 JETT PU SW	R 45	X1	LD	01	005	0 0 0 1	0.2 10K 600CC	1	2936
B00163B	14	341212	MACH SYNCHRO X	B00426C	62	LS	13	020	11.8 50	11.8 200	2	3161
B00164B	4	341231	CADC COARSE ALT X	B00110E	13	LS	13	010	11.8600	11.8 20	2	3267
B00164B	4	341231	CADC COARSE ALT X	B00722C	33	LS	13	010	11.8600	11.8 20	4	3269
B00163B	4	344116	CACS ALT X	B00109E	13	LS	12	010	11.8 50	11.8 200	1	4232
B00163B	4	344116	CACS ALT X	B00173E	13	LS	12	010	11.8 50	11.8 200	2	4238
B00164B	4	344116	CACS ALT X	B00110E	13	LS	12	010	11.8 50	11.8 200	1	4246
B00142A	X	344311	TILT SYNC STR X	B00138A	15	LS	13	20	11.8600	11.8 20	1	4449
B00142A	X	344311	TILT SYNC STR X	B00143A	SP	LS	13	20	11.8600	11.8 20	2	4450
B00109B	63	344311	ROLL NO. 3 X	B00143A	X	LS	13	20	11.8600	11.8 20	1	4462
B00109B	63	344311	ROLL NO. 3 X	B00426C	90	LS	13	20	11.8600	11.8 20	2	4463
B00142A	X	344312	TILT SYNC STR-X	B00140A	15	LS	13	20	11.8600	11.8 20	1	4554
B00142A	X	344312	TILT SYNC STR-X	B00143B	SP	LS	13	20	11.8600	11.8 20	2	4555
B0011CB	63	344312	ROLL NO. 3 X	B00143B	X	LS	13	20	11.8600	11.8 20	1	4567
B00143B	0	344312	IND A2 DATA STR-X	B00140B	22	LS	13	20	11.8600	11.8 20	2	4593
X 726	45	284112	TK 4 V-COMP HI Z	DB 324C	8	LZ	11	005	4.1M 233.5M C	4.1M.0073.5M C	1	2992
X 722	85	254112	TK 2 V-COMP HI Z	DB 324C	7	LZ	11	005	4.1M 233.5M C	4.1M.0073.5M C	1	2993
X 722	45	284112	TK 1 V-COMP HI Z	DB 324C	4	LZ	11	005	4.1M 233.5M C	4.1M.0073.5M C	1	2994
X 722	15	284112	R-TK 1 V-COMP HI Z	DB 324C	3	LZ	11	005	4.1M 233.5M C	4.1M.0073.5M C	1	2995
C 2684P	1	284112	SURGE TK HI Z	DB 324C	8	LZ	11	005	4.1M 133.5M C	4.1M.0043.5M C	1	2996
C 2662P	1	284112	SURGE TK HI Z	DB 324C	4	LZ	11	005	4.1M 133.5M C	4.1M.0043.5M C	1	2997

Figure B-7. Signal Mismatches by Terminal

AO SSIM MODULE COUNT, REDUNDANCY LEVEL 1									
SSIM TERM	1	2	3	4	5	6	7	8	TOTAL
1	4	0	0	0	0	0	0	0	4
2	2	0	0	0	0	0	0	0	2
3	2	0	1	0	0	0	0	0	3
4	0	1	1	0	0	0	0	0	2
5	3	1	3	0	0	0	0	0	7
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	1
9	2	1	3	0	0	0	0	0	6
10	8	1	1	0	0	0	0	0	10
11	1	0	0	0	0	0	0	0	1
12	0	0	0	0	0	0	0	0	0
13	1	2	0	0	0	0	0	0	3
14	0	2	1	0	0	0	0	0	3
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0
TOTAL	24	8	10	0	0	0	0	0	42

Figure B-8. Subsystem Interface Modules by SSIM Type

MODULE COUNT SUMMARY, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	7	64	9	15	75	26	24	0
2	54	0	5	0	0	34	8	0
3	0	0	0	0	0	0	10	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
TOTAL	61	64	14	15	75	60	42	0
TOTAL MODULES: 331								

Figure B-9. Summary of Subsystem Interface Modules

TERMINAL 1 MODULE COUNT, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	0	8	2	1	4	1	4	0
2	2	0	0	0	0	2	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	[illegible]
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
TOTAL	2	8	2	1	4	3	4	0
TOTAL MODULES: 24								

TERMINAL 2 MODULE COUNT, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	1	1	0	0	2	0	2	0
2	2	0	1	0	0	1	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
TOTAL	3	1	1	0	2	1	2	0
TOTAL MODULES: 10								

TERMINAL 3 MODULE COUNT, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	1	1	0	1	2	0	2	0
2	2	0	1	0	0	2	0	0
3	0	0	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
TOTAL	3	1	1	1	2	2	3	0
TOTAL MODULES: 13								

Figure B-10. Subsystem Interface Modules by Terminal

TERMINAL 1 UTILIZATION FACTOR, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	0.00	0.95	0.63	1.00	0.80	0.19	0.78	0.00
2	1.00	0.00	0.00	0.00	0.00	0.88	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TOTAL	1.00	0.95	0.63	1.00	0.80	0.65	0.78	0.00
TERMINAL UTILIZATION FACTOR: 0.86								

TERMINAL 2 UTILIZATION FACTOR, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	0.13	0.47	0.00	0.00	0.56	0.00	0.75	0.00
2	0.63	0.00	0.38	0.00	0.00	0.75	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TOTAL	0.46	0.47	0.38	0.00	0.56	0.75	0.75	0.00
TERMINAL UTILIZATION FACTOR: 0.54								

TERMINAL 3 UTILIZATION FACTOR, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	0.25	0.63	0.00	0.63	0.59	0.00	0.75	0.00
2	0.69	0.00	0.13	0.00	0.00	0.56	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TOTAL	0.54	0.63	0.13	0.63	0.59	0.56	0.67	0.00
TERMINAL UTILIZATION FACTOR: 0.55								

Figure B-11. Utilization Factor by Terminal

PDI SSIM UTILIZATION FACTOR, REDUNDANCY LEVEL 1									
SSIM TERMINAL	1	2	3	4	5	6	7	8	TOTAL
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
2	0.13	0.63	0.00	0.00	0.00	0.00	0.00	0.00	0.46
3	0.25	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.54
4	0.50	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.75
5	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
6	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
7	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
8	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
9	0.75	0.95	0.00	0.00	0.00	0.00	0.00	0.00	0.92
10	0.63	0.93	0.00	0.00	0.00	0.00	0.00	0.00	0.88
11	0.63	0.81	0.00	0.00	0.00	0.00	0.00	0.00	0.75
12	0.00	0.88	0.00	0.00	0.00	0.00	0.00	0.00	0.88
13	0.00	0.94	0.00	0.00	0.00	0.00	0.00	0.00	0.94
14	0.00	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.83
15	0.00	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.69
16	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50
17	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
18	0.00	0.75	0.00	0.00	0.00	0.00	0.00	0.00	0.75
19	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.50
TOTAL	0.50	0.83	0.00	0.00	0.00	0.00	0.00	0.00	0.79

Figure B-12. Utilization Factor by Subsystem Interface Module

UTILIZATION FACTOR SUMMARY, REDUNDANCY LEVEL 1								
SSIM	PDI	ADI	AI	SI	PDO	ADO	AO	SO
1	0.50	0.86	0.59	0.72	0.88	0.76	0.76	0.00
2	0.83	0.00	0.48	0.00	0.00	0.65	0.44	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.71	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TOTAL	0.79	0.86	0.55	0.72	0.88	0.70	0.68	0.00
OVERALL UTILIZATION FACTOR: 0.80								

Figure B-13. Summary Utilization Factor for System

PROGRAM: (CONUP)

ORIGIN REMOTE TERMINAL: 2

SIGNAL NAME	ID	ORIGIN	CONNECTOR	PINS	RT	DESTINATION	TYPE	UR
STE OR CLI DIV CMD	28049	8 2 0	0J0001/P3082	25	2	8 2 0	5	32
STE OR CLI DIV CMD	28050	8 2 0	0J0001/P3082	18	2	8 2 0	5	32
STEERING CMD	28061	8 2 0	0J0001/P0385	58	2	8 2 0	5	32
STEERING CMD	28062	8 2 0	0J0001/P0385	57	2	8 2 0	5	32
STEERING CMD	28066	8 2 0	0J0001/P3084	75	2	8 2 0	5	32
STEERING CMD	28067	8 2 0	0J0001/P3084	45	2	8 2 0	5	32
ELEVATION STEERING	28077	8 2 0	0J0001/P0385	61	2	8 2 0	5	32
AZIMUTH STEERING	28078	8 2 0	0J0001/P0385	60	2	8 2 0	5	32
AZIMUTH STEERING	28079	8 2 0	0J0001/P0385	59	2	8 2 0	5	32
LOCALIZER DEVIATION	28080	8 2 0	0J0001/P0385	56	2	8 2 0	5	32
LOCALIZER DEVIATION	28081	8 2 0	0J0001/P0385	55	2	8 2 0	5	32
VOLUME CONTROL	32035	8 2 0	1J0001/P	37 36	11	3 2 4	6	2
MAIN SQUELCH ADJ HI	39050	8 2 0	J 3/P	16	9	3 9 2	6	8
MAIN SQUELCH ADJ CT	39051	8 2 0	J 3/P	17	9	3 9 2	6	8
GUARD SQ ADJ HI	39052	8 2 0	J 3/P	10	9	3 9 2	6	8
GUARD SQ ADJ CT	39053	8 2 0	J 3/P	12	9	3 9 2	6	8
MAIN SQUELCH ADJ HI	39003	8 2 0	J0003/P	16	10	3 9 1	6	8
MAIN SQUELCH ADJ CT	39004	8 2 0	J0003/P	17	10	3 9 1	6	8
GUARD SQ ADJ HI	39005	8 2 0	J0003/P	10	10	3 9 1	6	8
GUARD SQ ADJ CT	39006	8 2 0	J0003/P	12	10	3 9 1	6	8
X DEFLECTION	43033	8 2 0	6J0001/P	M	12	4 3 6	6	32
Y DEFLECTION	43034	8 2 0	6J0001/P	A	12	4 3 6	6	32
AZ STAB (RTN)	21095	8 2 0	J0001/P	W	3	8 3 0	7	4
AZ STAB (SIN)	21093	8 2 0	J0001/P	S	3	8 3 0	7	16
AZ STAB (COS)	21094	8 2 0	J0001/P	R	3	8 3 0	7	16
ALTITUDE	21133	8 2 0	J000 /P6501	J	3	8 3 0	7	16
GRD RANGE	21134	8 2 0	J000 /P6501	D	3	8 3 0	7	16
SECTOR WIDTH SIN	21045	8 2 0	J0001/P	L	13	2 1 3	8	32
SECTOR WIDTH COS	21046	8 2 0	J0001/P	M	13	2 1 3	8	32
NO. 2 POINT BEARING	6001	8 2 0	0J0001/P3061	58 5 32	2	8 2 0	9	8
COMPASS X	46047	8 2 0	J0001/P	48	2	8 2 0	9	8
ORG RMT X	46050	8 2 0	J0001/P	35	2	8 2 0	9	8
COMPASS X	46086	8 2 0	J 1/P	48	3	8 3 0	9	8
BRG RMI X	46089	8 2 0	J 1/P	35	3	8 3 0	9	8
PILOT HSI (F)	24028	8 2 0	J000 /P	F	8	2 4 5	9	8
PILOT BDUI U	24119	8 2 0	J000 /P	R	8	2 4 5	9	8
TRUE AIRSPEED	26001	8 2 0	J /P		10	2 6 1	9	16
TILT CONTROL (X,Y,Z)	21016	8 2 0	J0001/P	D Q	13	2 1 5	9	16
STEERING ERROR +,-	27126	8 2 0	1J0006/P3076	81	2	8 2 0	9	32
STEERING ERROR +,-	27127	8 2 0	1J0006/P3076	82	2	8 2 0	9	32
TRUE GRD TRACK	27155	8 2 0	1J0004/P3074	56	2	8 2 0	9	32
TRUE GRD TRACK	27156	8 2 0	1J0004/P3074	55	2	8 2 0	9	32
RANGE TO DEST UNTS	27157	8 2 0	1J0004/P3074	74	2	8 2 0	9	32
RANGE TO DEST UNTS	27158	8 2 0	1J0004/P3074	73	2	8 2 0	9	32
RANGE TO DEST	27159	8 2 0	1J0004/P3074	93	2	8 2 0	9	32
RANGE TO DEST	27160	8 2 0	1J0004/P3074	92	2	8 2 0	9	32
RANGE TO DEST ILMS	27161	8 2 0	1J0004/P3074	77	2	8 2 0	9	32
RANGE TO DEST ILMS	27162	8 2 0	1J0004/P3074	76	2	8 2 0	9	32
RELATIVE TO DS	27163	8 2 0	1J0004/P3074	64	2	8 2 0	9	32
RELATIVE TO DS	27164	8 2 0	1J0004/P3074	63	2	8 2 0	9	32
RELATIVE TO DS	27165	8 2 0	1J0004/P3074	52	2	8 2 0	9	32

Figure B-14. Remote Terminal and Equipment Interconnect List

MODEL SA–T/T TRANSFER	
UPDATE RATE	BUS LOADING WORD/SEC
1	76
2	534
4	396
8	6,352
16	1,712
32	7,136
64	12,480
GRAND TOTAL	28,686

- BUS RESOURCE UTILIZATION 0.5737
- WORD LENGTH: 20
- BUS BIT RATE: 1000000

Figure B-15. Computation of Bus Loading and Utilization

MUXSIM BINARY MATRIX SCHEDULER

BUS SCHEDULE TABLE

U/R	IUR	NQN	MSLS	RTO	KRTO	KRTOM	URA	MSLD	RTA	KRTA	MSLDS	RTB	KRTB
64	1	6	195.0	0.2	1	1	64	195.0	0.2	1	195.0	0.2	1
32	2	28	223.0	0.4	1	1	32	223.0	0.4	1	223.0	0.4	1
16	4	13	107.0	0.2	1	1	16	107.0	0.2	1	504.0	0.9	1
8	8	98	794.0	1.4	2	2	16	397.0	0.7	1	0.0	0.0	1
4	16	14	99.0	0.2	1	1	4	99.0	0.2	1	99.0	0.2	1
2	32	36	267.0	0.5	1	1	2	267.0	0.5	1	267.0	0.5	1
1	64	11	76.0	0.1	1	1	1	76.0	0.1	1	76.0	0.1	1

BUS LOADING BY U/R

U/R = 64 KTROM = 1
GROUP 1

MSN	MSL
205	69.0
206	69.0
204	37.0
203	7.0
202	7.0
201	6.0
TOTAL	195.0

U/R = 32 KTROM = 1
GROUP 1

MSN	MSL
200	15.0
199	12.0
197	11.0
198	11.0
195	10.0
194	10.0
193	10.0
196	10.0
192	9.0
191	8.0
185	8.0
188	7.0
173	7.0
187	7.0
189	7.0
180	7.0
190	7.0

- BUS CAPACITY (BPS): 1000000
- WORD LENGTH (BPW): 20

Figure B-16. MUXSIM Bus Schedule and Loading

PROGRAM TMSULT—DATA BUS MESSAGE STRUCTURE LIST

	SIGNAL NAME	ID	TYPE	ORIGIN	DESTINATION
	PWR INTERLOCK OUT	46011	3	4 6 1	8 2 0
MESSAGE NO. 6 ORT: 10 DRT: 1 UR: 1					
	LDG GFAR DH POS-L1	1011	3	1 1 9	8 1 0
	LDG GFAR DR POS-L2	1012	3	1 1 9	8 1 0
	ANTISKID	10001	3	110 1	8 1 0
	BRAKE LOW PRESS	10002	3	110 2	8 1 0
MESSAGE NO. 7 ORT: 11 DRT: 12 UR: 1					
	LNG GEAR SW(AUSS)	43146	3	1 1 3	4 3 6
MESSAGE NO. 8 ORT: 12 DRT: 2 UR: 1					
	LAMP TEST GROUND	43015	3	4 3 6	8 2 0
	LAMP TEST GROUND	43016	3	4 3 6	8 2 0
	LAMP TEST GROUND	43017	3	4 3 6	8 2 0
	REC LEFT TURN LP G	43075	3	4 3 6	8 2 0
	REC SLOWDOWN LP GND	43076	3	4 3 6	8 2 0
	REC LETDOWN LP GND	43077	3	4 3 6	8 2 0
	REC EXECUTE LP GND	43078	3	4 3 6	8 2 0
	REC PULLUP LP GND	43079	3	4 3 6	8 2 0
	REC SPEED UP LP GND	43080	3	4 3 6	8 2 0
	REC RIGHT-TURN LP G	43081	3	4 3 6	8 2 0
	LEGEND DIM HI	43012	4	4 3 6	8 2 0
	MASTER LAMP GND	43070	4	4 3 6	8 2 0
	MASTER LOST LAMP GND	43071	4	4 3 6	8 2 0
	CAUTION LAMP GND	43072	4	4 3 6	8 2 0
	PROXIMITY WRMG LP G	43073	4	4 3 6	8 2 0
MESSAGE NO. 9 ORT: 13 DRT: 1 UR: 1					
	LDG GEAR DR POS-N1	1009	3	1 1 8	8 1 0
	LDG GEAR DR POS-N2	1010	3	1 1 8	8 1 0
MESSAGE NO. 10 ORT: 2 DRT: 12 UR: 1					
	LEGEND DIM	43031	5	8 2 0	4 3 6
	RANGE MARK URT CONT	43042	5	8 2 0	4 3 6
MESSAGE NO. 11 ORT: 12 DRT: 2 UR: 1					
	LEGEND DIM	43011	5	4 3 6	8 2 0
	LEGEND DIM	43053	5	4 3 6	8 2 0
	LEGEND DIM	43054	5	4 3 6	8 2 0
MESSAGE NO. 12 ORT: 2 DRT: 11 UR: 2					
	ANT CPLR BAND 1	32001	3	8 2 0	3 2 3
	ANT CPLR BAND 2	32002	3	8 2 0	3 2 3
	ANT CPLR BAND 3	32003	3	8 2 0	3 2 3
	ANT CPLR BAND 4	32004	3	8 2 0	3 2 3
	ANT CPLR BAND 5	32005	3	8 2 0	3 2 3
	ANT CPLR BAND 6	32006	3	8 2 0	3 2 3
	ANT CPLR BAND 7	32007	3	8 2 0	3 2 3
	ANT CPLR BAND 8	32008	3	8 2 0	3 2 3

Figure B-17. Data Bus Message Structure List

FUINO	MSLIOT	FRRAC
1	418.0	0.5350
2	697.0	0.8922
3	418.0	0.5350
4	195.0	0.2496
5	418.0	0.5350
6	701.0	0.8973
7	418.0	0.5350
8	294.0	0.3763
9	418.0	0.5350
10	697.0	0.8922
11	418.0	0.5350
12	195.0	0.2496
13	418.0	0.5350
14	701.0	0.8973
15	418.0	0.5350
16	462.0	0.5914
17	418.0	0.5350
18	697.0	0.8922
19	418.0	0.5350
20	195.0	0.2496
21	418.0	0.5350
22	701.0	0.8973
23	418.0	0.5350
24	294.0	0.3763
25	418.0	0.5350
26	697.0	0.8922
27	418.0	0.5350
28	195.0	0.2496
29	418.0	0.5350
30	701.0	0.8973
31	418.0	0.5350
32	271.0	0.3469
33	418.0	0.5350
34	697.0	0.8922
35	418.0	0.5350
36	195.0	0.2496
37	418.0	0.5350
38	701.0	0.8973
39	418.0	0.5350
40	294.0	0.3763
41	418.0	0.5350
42	697.0	0.8922
43	418.0	0.5350
44	195.0	0.2496
45	418.0	0.5350
46	701.0	0.8973
47	418.0	0.5350
48	462.0	0.5914
49	418.0	0.5350
50	697.0	0.8922
51	418.0	0.5350
52	195.0	0.2496
53	418.0	0.5350
54	701.0	0.8973
55	418.0	0.5350
56	294.0	0.3763
57	418.0	0.5350
58	697.0	0.8922

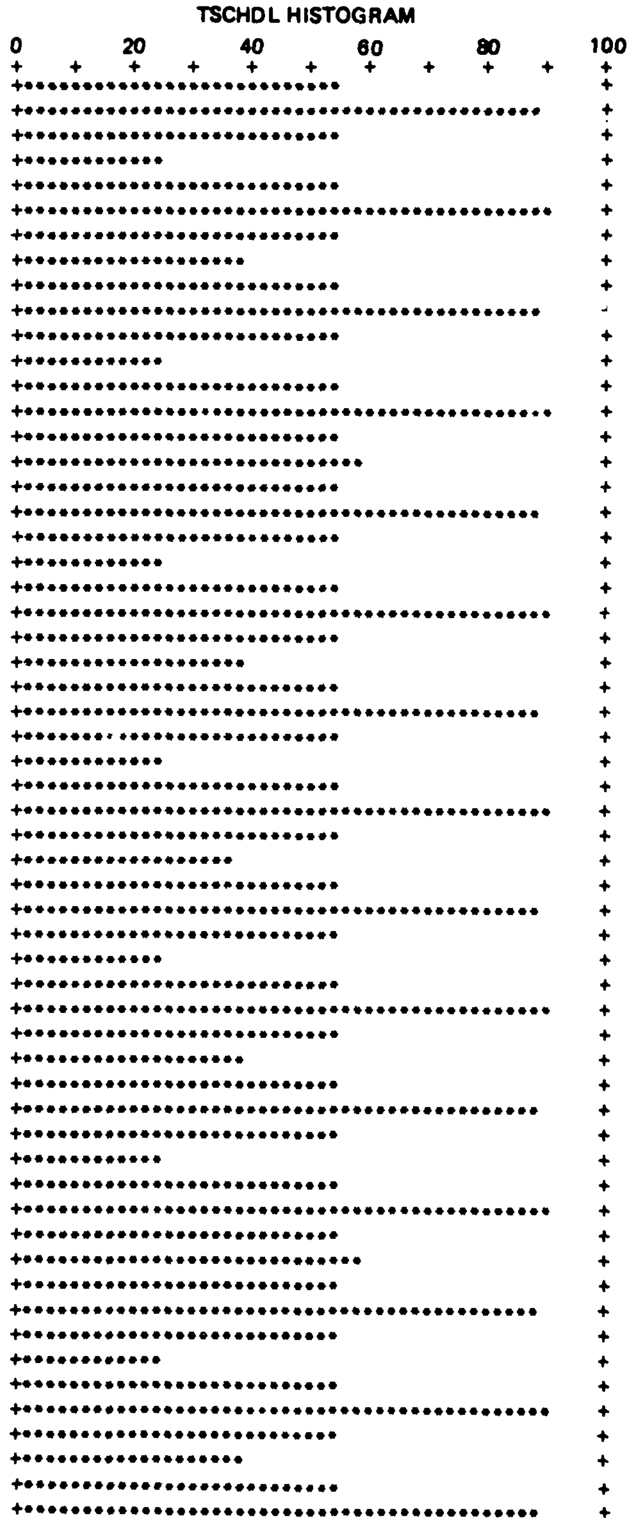


Figure B-18. Typical Minor Frame Loading

*** WORD FLOW SUMMARY LIST

MODEL SB-T/C/T TFR (BIT SHUFFLING)

ORT	DRT	OT	DT	U/R	QN	WDQ	WNS	SEL	CAT	SIGNAL ID
2	CEN	3	0	1	10	0	1	0	1	43013 43014 43027 43028 43029 43044 43045 43046 43047 43048
2	CEN	4	0	1	7	0	1	0	1	43019 43020 43021 43022 43023 43024 43025
2	CEN	5	0	1	2	0	2	0	2	43031 43042
5	CEN	3	0	1	1	0	1	0	1	10003
6	CEN	3	0	1	8	0	1	0	1	11009 11010 11011 11012 14015 14017 14019 14021
6	CEN	6	0	1	1	0	1	0	2	14009
6	CEN	8	0	1	1	0	1	0	2	14011
7	CEN	3	0	1	8	0	1	0	1	11013 11014 11015 11016 14016 14018 14020 14022
7	CEN	6	0	1	1	0	1	0	2	14010
7	CEN	8	0	1	1	0	1	0	2	14012
8	CEN	1	0	1	2	0	1	0	1	46009 46056
8	CEN	3	0	1	1	0	1	0	1	46011
10	CEN	3	0	1	4	0	1	0	1	1011 1012 10001 10002
11	CEN	3	0	1	1	0	1	0	1	43146
12	CEN	3	0	1	10	0	1	0	1	43015 43016 43017 43075 43076 43077 43078 43079 43080 43081
12	CEN	4	0	1	5	0	1	0	1	43012 43070 43071 43072 43073
12	CEN	5	0	1	3	0	3	0	2	43011 43053 43054
13	CEN	3	0	1	2	0	1	0	1	1009 1010
CEN	1	3	0	1	20	0	0	0	1	1009 1010 1011 1012 10001 10002 10003 11009 11010 11011 11012 11013 11014 11015 11016 14015 14017 14019 14021 14016
CEN	1	3	0	1	3	0	2	0	1	14018 14020 14022
CEN	1	6	0	1	2	0	2	0	2	14009 14010
CEN	1	8	0	1	2	0	2	0	2	14011 14012
CEN	2	1	0	1	2	0	1	0	1	46009 46056
CEN	2	3	0	1	11	0	1	0	1	43015 43016 43017 43075 43076 43077 43078 43079 43080 43081 46011
CEN	2	4	0	1	5	0	1	0	1	43012 43070 43071 43072 43073
CEN	2	5	0	1	3	0	3	0	2	43011 43053 43054
CEN	12	3	0	1	11	0	1	0	1	43146 43013 43014 43027 43028 43029 43044 43045 43046 43047 43048
CEN	12	4	0	1	7	0	1	0	1	43019 43020 43021 43022 43023 43024 43025
CEN	12	5	0	1	2	0	2	0	2	43031 43042
1	CEN	1	0	2	2	0	1	0	1	25009 25010
2	CEN	1	0	2	20	0	0	0	1	34001 34002 34003 34004 34005 34006 34007 34008 34009 34010 34011 34012 34013 34014 34015 34016 34017 34018 34019 34020
2	CEN	1	0	2	1	0	2	0	1	34021
2	CEN	2	[illegible]	2	2	0	1	0	1	32031 32037
2	CEN	3	[illegible]	2	20	0	0	0	1	21054 21066 21067 21068 32001 32002 32003 32004 32005 32006 32007 32008 32009 32010 32011 32012 32013 32014 32015 32016
2	CEN	3	0	2	20	0	0	0	1	32017 32018 32019 32020 32021 32038 32040 32041 34022 34023 39017 39018 39019 39064 39065 39066 41008 41009 41010 41011
2	CEN	3	0	2	8	0	3	0	1	41012 41013 41042 41043 41044 41045 41046 41047
2	CEN	4	0	2	12	0	1	0	1	21020 32022 34024 43087 43089 43091 43093 43095 43097 43099 47007 47025
2	CEN	5	0	2	5	0	5	0	2	32028 32029 32030 32033 32034

Figure B-19. Message Flow Summary

*** MESSAGE FLOW SUMMARY LIST

MODEL SB-T/C/T TFR (RIT SHUFFLING)

ORT	DRT	OT	DT	U/R	ON	WDL	MSL	BSL	CAT	MSN	SIGNAL ID
CEN	1	0	0	1	20	0	2.5	2.5	0	1	1009 1010 1011 1012 10001 10002 10003 11009 11010 11011 11012 11013 11014 11015 11016 14015 14017 14019 14021 14016
2	CEN	0	0	1	17	2	4.5	4.5	0	2	43013 43014 43027 43028 43029 43044 43045 43046 43047 43048 43019 43020 43021 43022 43023 43024 43025
5	CEN	0	0	1	1	1	3.5	3.5	0	3	10003
6	CEN	0	0	1	10	3	5.5	5.5	0	4	11009 11010 11011 11012 14016 14017 14019 14021 14009 14011
7	CEN	0	0	1	10	3	5.5	5.5	0	5	11013 11014 11015 11016 14016 14018 14020 14022 14010 14012
8	CEN	0	0	1	3	2	4.5	4.5	0	6	46009 46056 46011
10	CEN	0	0	1	4	1	3.5	3.5	0	7	1011 1012 10001 10002
11	CEN	0	0	1	1	1	3.5	3.5	0	8	43146
12	CEN	0	0	1	15	2	4.5	4.5	0	9	43015 43016 43017 43075 43076 43077 43078 43079 43080 43081 43012 43070 43071 43072 43073
13	CEN	0	0	1	2	1	3.5	3.5	0	10	1009 1010
CEN	12	0	0	1	18	2	4.5	4.5	0	11	43146 43013 43014 43027 43028 43029 43044 43045 43046 43047 43048 43019 43020 43021 43022 43023 43024 43025
CEN	2	0	0	1	18	3	5.5	5.5	0	12	46009 46056 43015 43016 43017 43075 43076 43077 43078 43079 43080 43081 46011 43012 43070 43071 43072 43073
2	CEN	0	0	1	2	2	4.5	4.5	0	13	43031 43042
CEN	12	0	0	1	2	2	4.5	4.5	0	14	43031 43042
CEN	1	0	0	1	7	6	8.5	8.5	0	15	14018 14020 14022 14009 14010 14011 14012
12	CEN	0	0	1	3	3	5.5	5.5	0	16	43011 43053 43054
CEN	2	0	0	1	3	3	5.5	5.5	0	17	43011 43053 43054
2	CEN	0	0	2	20	0	2.5	5.0	0	18	34001 34002 34003 34004 34005 34006 34007 34008 34009 34010 34011 34012 34013 34014 34015 34016 34017 34018 34019 34020
2	CEN	0	0	2	20	0	2.5	5.0	0	18	21054 21066 21067 21068 32001 32002 32003 32004 32005 32006 32007 32008 32009 32010 32011 32012 32013 32014 32015 32016
2	CEN	0	0	2	20	0	2.5	5.0	0	18	32017 32018 32019 32020 32021 32038 32040 32041 34022 34023 39017 39018 39019 39064 39065 39066 41008 41009 41010 41011
3	CEN	0	0	2	20	0	2.5	5.0	0	19	43105 43106 43107 43108 43109 43110 43113 43114 43118 43119 43120 43121 43122 43123 43124 43125 43126 43127 43128 43129
10	CEN	0	0	2	20	0	2.5	5.0	0	20	53003 53004 53005 53006 53007 53061 53062 53064 53066 53079 53060 53065 53067 53068 53069 53070 53071 53073 53074 53075
CEN	12	0	0	2	20	0	2.5	5.0	0	21	43087 43089 43091 43093 43095 43097 43099 43105 43106 43107 43108 43109 43110 43113 43114 43118 43119 43120 43121 43122
CEN	11	0	0	2	20	0	2.5	5.0	0	22	32001 32002 32003 32004 32005 32006 32007 32008 32009 32010 32011 32012 32013 32014 32015 32016 32017 32018 32019 32020

Figure B-20. Word Flow Summary

APPENDIX C

BUS NETWORK MODELING

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

APPENDIX C

BUS NETWORK MODELING

1.0 BUS NETWORK MODELING

Another important design activity that must be completed early in avionics system integration is verification of waveform integrity at the receiver of each terminal on the 1553 data bus. This is usually done initially by computer models. This section contains two model examples and a brief discussion of the need for models.

2.0 NEED FOR MODELS

The analog transmit-receive function of a 1553 terminal is the interface of the digital logic with the data bus. Even though the transmitted signal on the bus is generated in digital form, the twisted-shielded pair transmission line characteristics, along with multiple terminations, cause the signal that is received by a terminal to be attenuated. A 1553 data bus is terminated at each end in the cable characteristic impedance to minimize reflections. When the stubs are added for connection of the terminals, the bus is loaded locally and a mismatch occurs with resulting reflections. The degree of mismatch and signal distortions caused by reflections are a function of the impedance presented by the stub and terminal input impedance. To minimize signal distortion, it is desirable to maintain a high stub impedance reflected back to the main bus. At the same time, the impedance needs to be kept low so that adequate signal power will be delivered to the receiver input. Trade-off and compromise between these conflicting requirements are necessary to achieve the specified signal-to-noise ratio and system error performance.

A system design consideration is the definition of the bus network and specification of the terminal interfaces. The bus network must be designed for signal integrity to achieve bit error and word error rate performance required by 1553. Error rate is a function of the system network configuration as well as bus interface hardware designs. Unfortunately, bus error rate performance is not easily predicted for a specific bus network because of the many variables affecting the quality of a waveform from the time it is transmitted to the time it is received. These factors can be treated separately: (1) factors affecting waveform integrity at any given stub and (2) characteristics of the receiver affecting data reception at that stub.

MIL-STD-1553 has attempted to specify the more obvious characteristics that affect error rate, but difficulty arises when one tries to specify such characteristics as the type of receiver filter or sync detect algorithm required. Dictating implementation of design cannot be the intent of the standard. Instead, receiver interface requirements, rather than design, have been devised to "guarantee" proper bus performance.

The multiplex system should be designed to work successfully with up to 32 terminals (i.e., 31 remote terminals and 1 monitor), and stub lengths should not exceed 20 ft. Computer simulations are usually used to develop parametric data that may be used to define the waveform distortion to be expected from a given configuration. One technique commonly used is to reduce these data to envelopes of permissible cumulative stub and trunk lengths. Separate cases must be developed for bus configurations employing transformer stub couplers and those that do not.

Most of the factors that affect waveform integrity are as follows:

a. Transmitter dynamic output impedance

b. Transmitter waveform symmetry

c. Transmitter rise and fall times

d. Bus trunk line length

e. Bus stub lengths and placement on the trunk

f. Bus trunk terminating impedance

g. Complex impedance (R, L, C) of all receivers on line

h. Complex impedance and transfer characteristics of coupling transformer (if used)

i. Injected noise

The following factors affect optimal receiver operation:

a. Type of receiver filter (if any)
b. Threshold levels of receiver comparator(s)
c. Ability of synchronizer to accept distorted data
d. Algorithm for sync detection

It is usually appropriate to vary the factors of the receiver in separate simulation runs from variations in the bus layout factors.

Many different 1553 bus configurations can be built that meet the error rate requirements of 1553, but careful attention must be given to the number, length, and location of the stubs on the main bus. It is usually necessary to verify adequacy of the layout using a computer simulation or laboratory test setup. The computer-generated data bus simulation is a technique that has more user flexibility, which is desirable during early system design. A number of bus network simulation programs have been developed with varying degrees of success.

3.0 SIMULATION MODELS

3.1 EXAMPLE ONE: AVIONICS MULTIPLEX SUBSYSTEM SIMULATION

The avionics multiplex (AMUX) subsystem simulation program package was developed during the B-1 program (USAF contract F33657-72-C-0600) and was designed to derive a time-dependent response of a particular AMUX configuration. The algorithm used is a time domain-impulse analysis technique that was initially run on a Hewlett-Packard 9820 desk calculator.

This AMUX simulation program is written in BASIC and is run under the RST-E operating system on a DEC-PDP/11. The program uses an interactive approach where the user is asked for various inputs, as opposed to a batch system in which no interaction occurs. The output of the program is used to produce graphs and summary statistics of the run; figures C-1 and C-2 show examples.

The package provides input of a particular line configuration and capability to change any input parameter. The total simulation process, from input of a line configuration to completion of the graphs, takes an average of 1 hr. However, simulation and plotting may be run at two different times.

3.1.1 Line Simulation

The line simulation is modeled as shown in figure C-3. The line is represented as a large array. A piece of wire or transmission line is modeled with the elements of the array as shown in figure C-4.

An impulse, or the first derivative of a step function, is stepped from cell to cell in the top set of elements to represent the transmitted or incident signal and is stepped back through the second set of elements in the opposite direction to represent the reflected signal. Each step from one cell to the next cell represents the movement of a signal impulse along a small section of transmission line. The length of the line represented by each cell is specified by the user (default value is 4 ft) and is called the resolution. The increment of time that is taken in the shift (dt) is a function of this resolution and the propagation delay associated with the particular wire used. Figure C-5 shows how a line with stubs is represented by the array.

Use of the first derivative of the signal facilitates the calculations involved in propagating the signal through the line. Various coefficients are used to describe the effects of the various components of the line (see fig. C-6).

a. Transmission: main-main

$$CTMM = (Z_o + Z_s)/(1.5\ Z_o + Z_s)$$

b. Transmission: main-stub, stub-main

$$CTMS = Z_o/(1.5Z_o + Z_s)$$

c. Reflection: main

$$CRM = -0.5Z_o/(1.5Z_o + Z_s)$$

d. Reflection: stub

$$CRS = (Z_s - 0.5Z_o)/(1.5Z_o + Z_s)$$

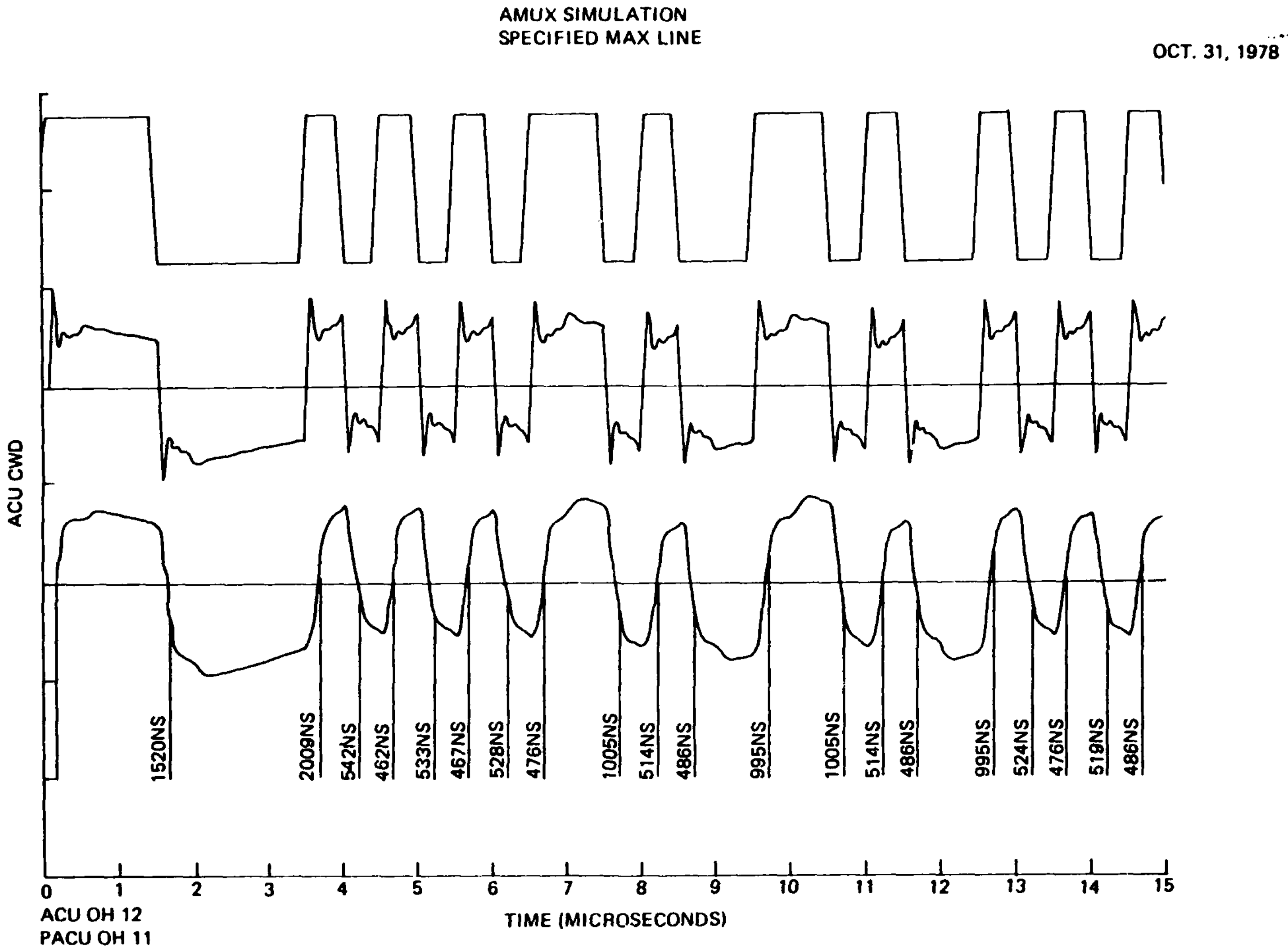


Figure C-1. Sample Graphic Output

AMUX SIMULATION V5 5

PULSE MAX & MIN

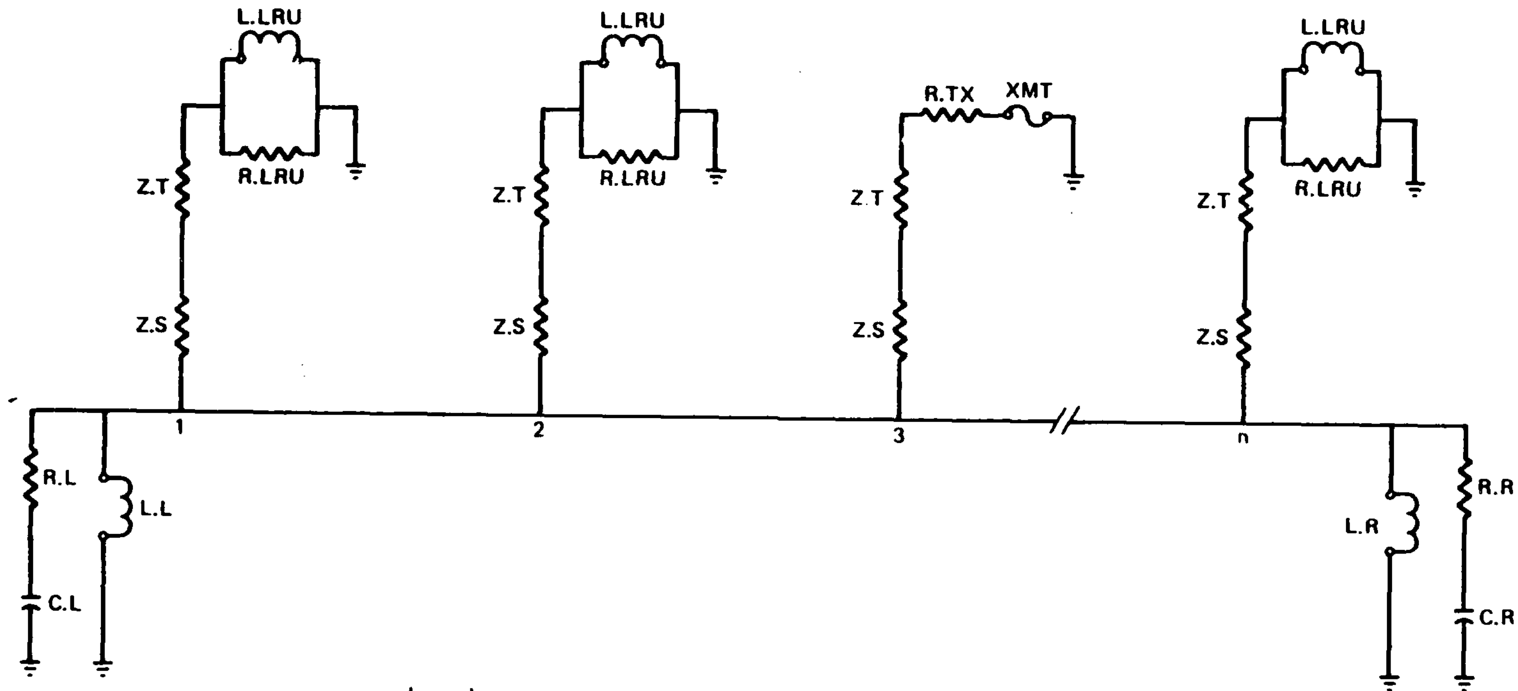
FILE AMUX52 31-OCT-78

PULSE DURATION	MAX	MIN
500 NS	542 NS	462 NS
1000 NS	1005 NS	995 NS
1500 NS	1528 NS	– – –
2000 NS	2009 NS	– – –

——— NO MAX OR MIN FOUND

– – – PULSE WIDTH NOT USED

Figure C-2. Data Summary Statistics Output



Legend:

Z0	line impedance	R.L	left-end resistance
Z.S	impedance of stub at node	C.L	left-end capacitance
Z.T	terminal impedance	L.L	left-end inductance
L.LRU	LRU inductance	R.R	right-end resistance
R.LRU	LRU resistance	C.R	right-end capacitance
R.TX	transmitter resistance	L.R	right-end inductance
#	stub number		

Figure C-3. Line Impedance Model

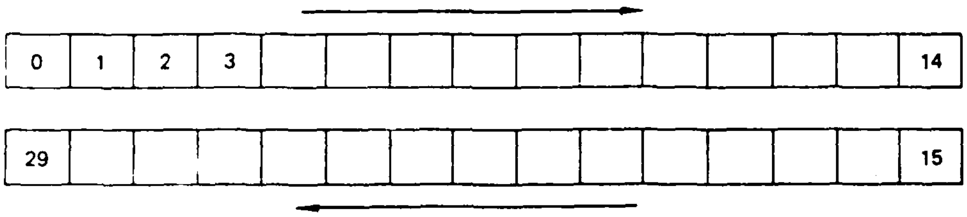


Figure C-4. Transmission Line Array Representation

e. Transmission: transmitter-stub

$$CTT = (Z_o + Z_t)/(R_{tx} + Z_o + Z_t)$$

f. Reflection: transmitter

$$CRT = (R_{tx} + Z_t - Z_o)/(R_{tx} + Z_t + Z_o)$$

The coefficients modify the signal on the line according to the algorithm shown in figure C-6.

For the line replaceable units (LRU), the modified signal is:

Given the following:

Vr = first derivative of the reflected voltage
Vs = first derivative of the incident voltage

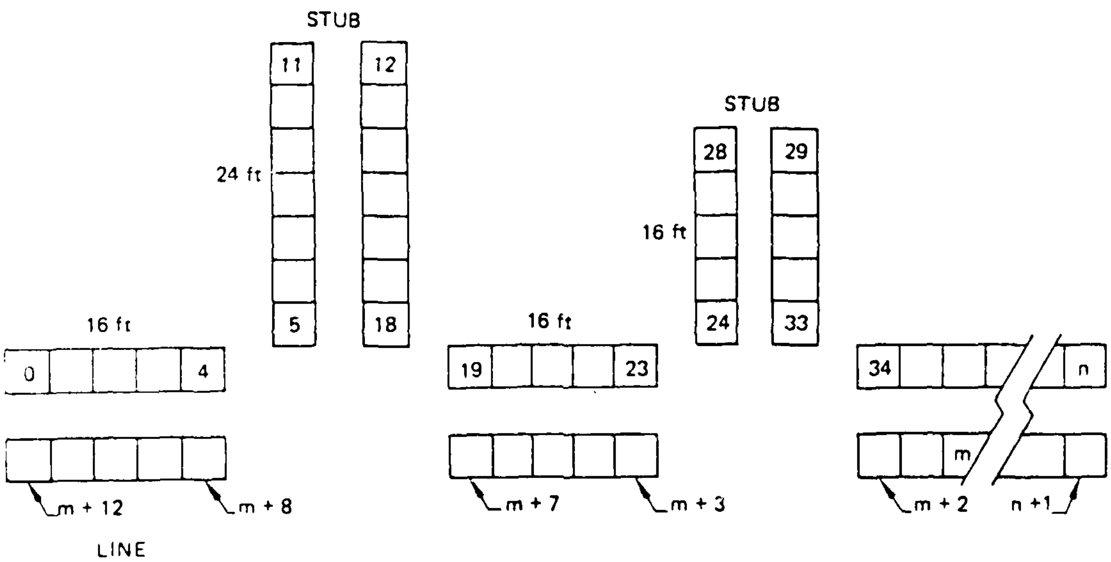


Figure C-5. Line With Stubs Array Presentation

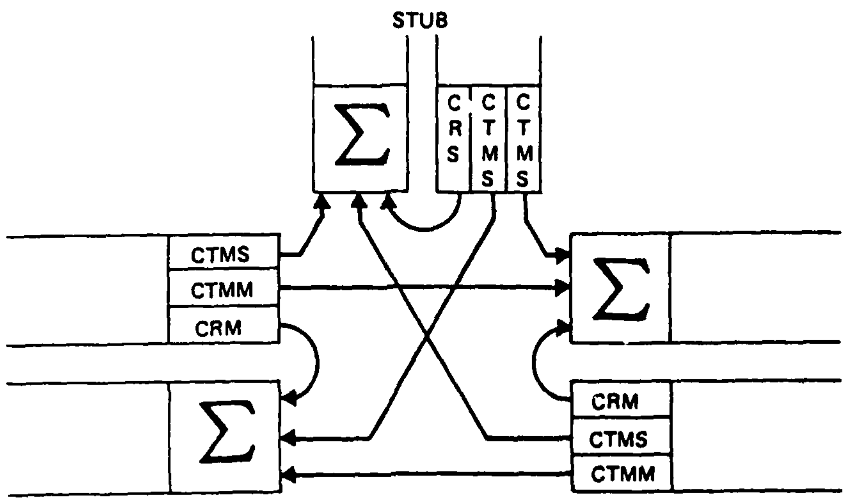


Figure C-6. Line and Stub Effects Nomenclature and Algorithm

Effects due to resistance:

$$CR = (R_{lru} - Z_o + Z_t)/(R_{lru} + Z_o + Z_t)$$

Effects of the inductance:

$$CL = \frac{dt\ (Z_o + Z_t)R_{lru}}{(Z_o + Z_t + R_{lru})L_{lru}} \sum_{o}^{t} (V_s + V_r)$$

The overall effect is Vr = VsCR - CL

The combined effects of shunt inductors and series capacitors at the line end terminations are shown in figure C-7.

The processes that occur during one time frame are:

a. Contents of the line array are shifted to the next higher index.
b. Signals resulting from the stub junctions are calculated.
c. Calculations are performed for the LRUs and line terminations.
d. One time frame of the message is transmitted.

The output of the line is taken from the Σ(Vs + Vr) term of the receiver LRU calculation. This output is then run through a filter model to filter the output (see sec. 3.1.2).

The following restrictions apply to this simulation:

a. The maximum total main line length is 600 ft, maximum total stub length is 300 ft.

b. There is a maximum of 32 stubs on the multiplex line.

c. Multiple stubs at one node have not been modeled. If the line being simulated has multiple stubs, separate the stubs by a distance equal to one time frame.

d. Connectors between two wire segments cannot be modeled in this version.

e. Only one LRU configuration is allowed on a stub.

f. There is a maximum of 20 different LRU configurations.

g. All wire lengths are rounded to the nearest nonzero multiple of the resolution.

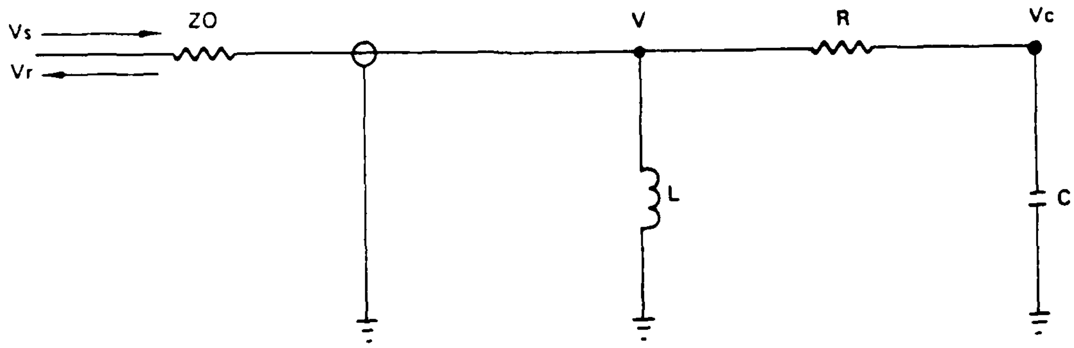
h. Because of computer memory space restrictions, limits must be placed on transmission time and resolution for a given propagation velocity (table C-1).

i. The filter used in this version is the Boeing Modem III design. A future version will allow user specification of a filter.

3.1.2 Filter Simulation

The potential time-domain model assumptions and derivation are shown in figure C-8.

Examples of an input waveform and a filtered output (waveform at the output of a terminal's filter) are shown in figures C-9 and C-10.



Given that: $V = \sum_{0}^{t} (Vs + Vr), \; Vc = \sum_{0}^{t} \frac{dt}{RC} (V - Vc)$

then $Vr = Vs \frac{R - Z_0}{R - Z_0} \frac{dt}{L} V \frac{RZ_0}{R + Z_0} + \frac{dt}{RC} (V - Vc) \frac{Z_0}{R + Z_0}$

Figure C-7. Line Termination Electrical Representation

Table C-1. Transmission Time and Line Resolution Tradeoff

RUNNH
ANALYSIS OF UPPER LIMITS TO AMUX SIMULATION

GIVEN THAT THE MAXIMUM NUMBER OF TIME FRAMES OF WIDTH DT IS 6000 AND THE MAXIMUM TRANSMISSION TIME IS 64 MICROSECONDS, THE FOLLOWING PARAMETERS ARE VARIABLE:

RESOLUTION & PROPAGATION VELOCITY.

THE MAXIMUM ALLOWABLE TRANSMISSION TIME IS THEN COMPUTED.

VARYING THE RESOLUTION FROM 1' TO 10' BY .5' INCREMENTS

PROPAGATION VELOCITY = .5 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.203252E-8	12.1951	246
1.5	.304878E-8	18.2927	164
2	.406504E-8	24.3902	123
2.5	.510204E-8	30.6122	98
3	.609756E-8	36.5854	82
3.5	.714286E-8	42.8571	70
4	.819672E-8	49.1803	61
4.5	.909091E-8	54.5455	55
5	.102041E-7	61.2245	49

PROPAGATION VELOCITY = .55 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.185185E-8	11.1111	270
1.5	.277778E-8	16.6667	180
2	.37037E-8	22.2222	135
2.5	.462963E-8	27.7778	108
3	.555556E-8	33.3333	90
3.5	.649351E-8	38.961	77
4	.735294E-8	44.1176	68
4.5	.833333E-8	50	60
5	.925926E-8	55.5556	54
5.5	.102041E-7	61.2245	49

Table C-1. Transmission Time and Line Resolution Tradeoff (Continued)

PROPAGATION VELOCITY = .6 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.169492E-8	10.1695	295
1.5	.255102E-8	15.3061	196
2	.340136E-8	20.4082	147
2.5	.423729E-8	25.4237	118
3	.510204E-8	30.6122	98
3.5	.595238E-8	35.7143	84
4	.675676E-8	40.5405	74
4.5	.769231E-8	46.1538	65
5	.847458E-8	50.8475	59
5.5	.925926E-8	55.5556	54
6	.102041E-7	61.2245	49

PROPAGATION VELOCITY = .65 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.15674E-8	9.40439	319
1.5	.234742E-8	14.0845	213
2	.3125E-8	18.75	160
2.5	.390625E-8	23.4375	128
3	.471698E-8	28.3019	106
3.5	.549451E-8	32.967	91
4	.625E-8	37.5	80
4.5	.704225E-8	42.2535	71
5	.78125E-8	46.875	64
5.5	.862069E-8	51.7241	58
6	.943396E-8	56.6038	53
6.5	.102041E-7	61.2245	49

PROPAGATION VELOCITY = .7 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.145349E-8	8.72093	344
1.5	.218341E-8	13.1004	229
2	.290698E-8	17.4419	172
2.5	.364964E-8	21.8978	137
3	.434783E-8	26.087	115
3.5	.510204E-8	30.6122	98
4	.581395E-8	34.8837	86
4.5	.657895E-8	39.4737	76
5	.724638E-8	43.4783	69
5.5	.806452E-8	48.3871	62
6	.877193E-8	52.6316	57
6.5	.943396E-8	56.6038	53
7	.102041E-7	61.2245	49

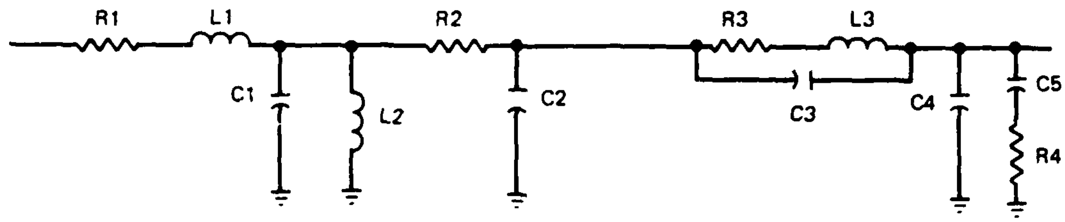
Table C-1. Transmission Time and Line Resolution Tradeoff (Concluded)

PROPAGATION VELOCITY = .75 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.13587E-8	8.15217	368
1.5	.203252E-8	12.1951	246
2	.271739E-8	16.3043	184
2.5	.340136E-8	20.4082	147
3	.406504E-8	24.3902	123
3.5	.47619E-8	28.5714	105
4	.543478E-8	32.6087	92
4.5	.609756E-8	36.5854	82
5	.675676E-8	40.5405	74
5.5	.746269E-8	44.7761	67
6	.819672E-8	49.1803	61
6.5	.877193E-8	52.6316	57
7	.943396E-8	56.6038	53
7.5	.102041E-7	61.2245	49

PROPAGATION VELOCITY = .8 * C

RESOLUTION	DT (SEC)	TRANSMISSION TIME (MSEC)	TIME FRAMES PER 500NS
1	.127226E-8	7.63359	393
1.5	.19084E-8	11.4504	262
2	.255102E-8	15.3061	196
2.5	.318471E-8	19.1083	157
3	.381679E-8	22.9008	131
3.5	.446429E-8	26.7857	112
4	.510204E-8	30.6122	98
4.5	.574713E-8	34.4828	87
5	.632911E-8	37.9747	79
5.5	.704225E-8	42.2535	71
6	.769231E-8	46.1538	65
6.5	.833333E-8	50	60
7	.892857E-8	53.5714	56
7.5	.961538E-8	57.6923	52
8	.102041E-7	61.2245	49



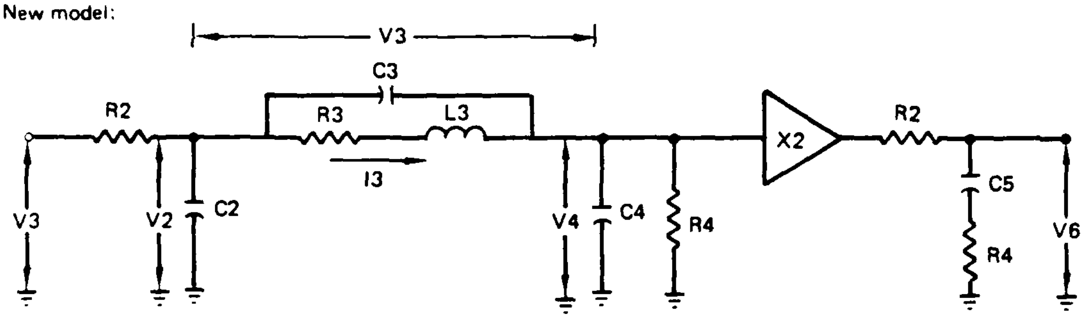
Typical values:

R1 = 102 ohms (RS)	L1 = L_{dist}	C1 = 100 pF C_{dist}
R2 = 1,210 ohms	L2 = 10 mH L_{pri}	C2 = 22 pF
R3 = 12 ohms (1/Q)	L3 = 100 μH	C3 = 7 pF C_{dist}
R4 = 1,210 ohms		C4 = 150 pF
		C5 = 22 nF

Model assumptions:

- Delete L2 and R1, provided by line model
- Delete L1 and C1, not significant to line modeling
- Separate the effects of C5 from others

New model:



Initialization:

K1 = dt/(C2 + C3)	V2 = 0
K2 = dt/C4	V4 = 0
K3 = dt/L3	V5 = 0
K4 = dt/2R4C5	I3 = 0
K5 = C3/C4	

The calculations performed each time frame are:

$$V2 = V2 + K1\left((V1 - V2)/R2 - I3\right)$$
$$I3 = I3 + K3(V2 - I3R3 - V4)$$
$$V4 = V4 + K2(I3 - V4/R4)$$
$$V5 = V5 + K4(2V4 - V5)$$
$$V6 = V4 + V5/4$$

The output of the filter is taken as V6.

Figure C-8. Potential Time Domain Model

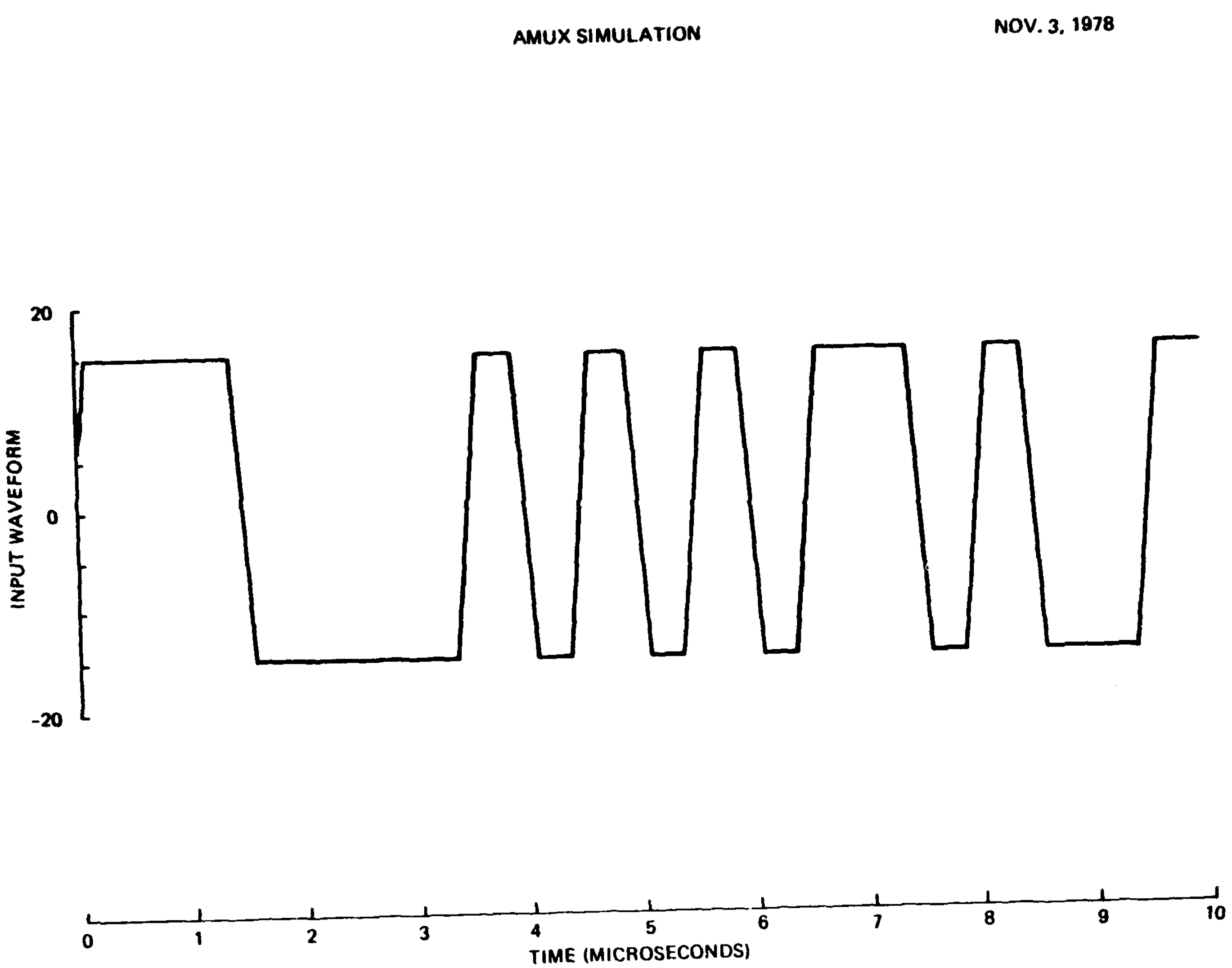


Figure C-9. Example of Input Waveform

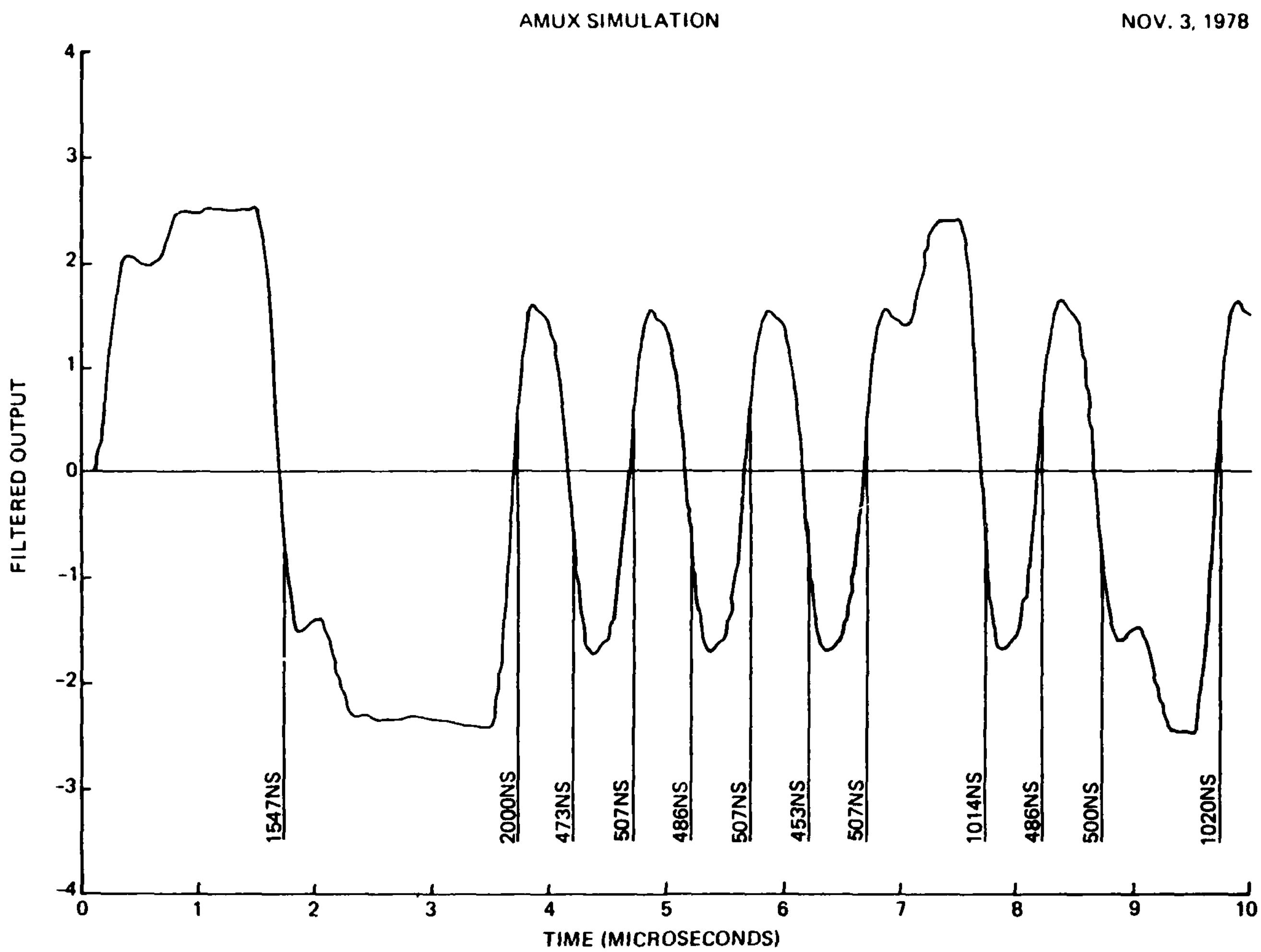


Figure C-10. Example of Filtered Output Waveform

3.2 EXAMPLE TWO: AIR FORCE AVIONICS LABORATORY DATA BUS NETWORK SIMULATION

The Air Force Avionics Laboratory (AFAL) sponsored a simulation program and related verification hardware, and this effort was reported in AFAL Technical Report AFAL-TR-75-209 "Data Bus Network Simulation." The following four paragraphs summarize this report.

This report describes a general-purpose digital computer program that provides the capability to aid in design and evaluation of complex cable networks. Programming routines are coded in Fortran IV for easy modification and for operation on either an IBM System 370 or a Digital Equipment Corporation DEC-10 computer. The simulator is specifically designed to emulate 1 M MIL-STD-1553-type waveforms and bus configurations.

A typical MIL-STD-1553 data bus consists of a main bus accessed through shorter cables called stubs. These stubs present a capacitive load to the bus coupler that is located at the main bus, stub junction. Coupler transformers with their stub loads are simulated as second-order systems with parameters derived from measurement data. Tustin's transformation is used to produce different equations that permit time-domain computation. Iterative time-domain calculations are employed. Reflection coefficients are used to calculate signal imperfections caused by junction and termination discontinuities. Transmission line filter effects are accommodated by an independent algorithm that takes skin effects into account. Procedures are included for operating the program and characterizing the line and transformer couplers from laboratory data. The program will permit stubs of lengths up to nominally 20 ft at numerous points along the main bus trunk. Multiple stubs can also be simulated so that the amount of stub loading is not bounded.

To verify validity of simulation, a breadboard of a typical bus network was constructed. This breadboard permitted a worst case configuration of a 300 ft main bus trunk with eight stubs connected via either of two types of transformer couplers. Results produced by the simulation agreed favorably with data taken on the breadboard. An additional transformer and stub characterization study is recommended to better correlate transformer parameters to the software model.

The report includes all equations and the Fortran source code.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

APPENDIX D

LIFE CYCLE COST ANALYSIS

1.0 INTRODUCTION

The life cycle costs (LCC) of a multiplex system embrace both acquisition costs and operating and support (O&S) costs. LCC analyses are performed early in the acquisition phase to help the designer make decisions on configuration selection and system performance. These analyses also help identify high-cost system elements that merit special management and design attention. LCC analysts monitor and refine original estimates, evaluate proposed and imposed design changes, and augment technical trade studies at finer levels of detail.

Cost models are generally used to make LCC estimates. The three model types are forecast sensitivity or parametric, cost-estimating relationships (CER), and accounting. A forecast sensitivity model uses equations developed from analysis of past experience (e.g., the RCA PRICE model). It can predict cost sensitivity to changes in schedule, parts, and quality. A CER is an equation that predicts cost in terms of performance or design parameters (e.g., mean time between failures, weight, data rate). An accounting model is a set of equations that expresses the cost in terms of the estimated actual labor or materials expended (e.g., design cost as a function of the various man-hours budgeted for circuit design, breadboard fabrication, laboratory test, etc.). LCC estimates are usually performed by LCC personnel, as they are most cognizant of applicable models. These specialists, who respond to program office and contract requirements, must be thoroughly briefed on multiplex system characteristics to intelligently choose models. LCC analysis is therefore a team effort of LCC model specialists and project engineers. The system engineer must inform the analyst of the design trade-offs that must be considered so that LCC personnel can evaluate potential model capability to support these trades. Inputs to the LCC models are supplied by Engineering, Finance, and Logistic Support organizations depending on cost model requirements. Complex cost models may be computerized; therefore programming support may also be required.

This section of the handbook describes cost elements in the life cycle, discusses cost models in terms of both sources and several specific models that may be helpful, presents a reliability prediction model for airborne multiplex systems, and discusses cost versus MTBF optimization.

2.0 COST ELEMENTS IN THE LIFE CYCLE

2.1 COST DRIVERS AND HIGH-VALUE ITEMS

A cost driver is a design or performance requirement that significantly influences the cost of a system. For example, a part quality specification, a requirement for 100% testing of all production components and subassemblies, and a very high mean time between failure (MTBF) performance requirement could be cost drivers. After the initial LCC estimate is made for a multiplex system, analysis should be performed to identify the cost drivers. This type of analysis is conducted by asking "what if" type of questions to Finance and Engineering personnel. For example: What if JAN parts could be used instead of hi-rel parts? What if the error detection probability requirement was relaxed? What if central rather than distributed processing is required? Cost driver identification is important because it allows program design emphasis to be intelligently applied, alerts management to critical areas, and provides a rational basis for proposing changes in requirements.

A high-value item is a unit of hardware or software whose total cost is large compared with that of other units. A connection or circuit board, although of low unit cost, may be used in the system in sufficient quantity to become a high-value item. High-value items should be identified after the initial LCC estimate is made to enable cost reduction studies to be defined and to alert engineers and management to the presence of these key items. In most designs, less than 20% of the unique types of hardware items constitutes over 80% of the total production cost. It is also generally found that most of the high-value production items are also high-value O&S items. A representative high-value item chart is shown in figure D-1. This is sometimes called a Pareto chart. The items are listed in descending cost order. Remember to include total quantity of each item in the production run and take learning curves into account (there are different production learning curves for different types of electronic components).

2.2 DEFINITION OF ACQUISITION AND O&S COSTS

Acquisition cost elements include all deliverable items and the tasks and test items that result in deliverables. Deliverables include operational hardware, operational software, training equipment, support equipment (e.g., special testers) not already in the inventory, support software, initial spares, and all operating and maintenance manuals that are needed at bases and repair depots. The tasks to be costed include concept definition and preliminary design, detailed engineering, in-house laboratory and field test planning, test programs, cadre training, production, production testing of complete systems and other deliverables, delivery (if required), installation, and checkout.

O&S cost elements include replenishment spares at base and depot levels, inservice training of maintenance personnel, pay and allowances for all support personnel assigned to maintenance for both hardware and software, and initial and annual data management costs for new items in the Federal stock catalog. Since a multiplex system consumes little energy, operating costs are expected to be negligible compared with support costs.

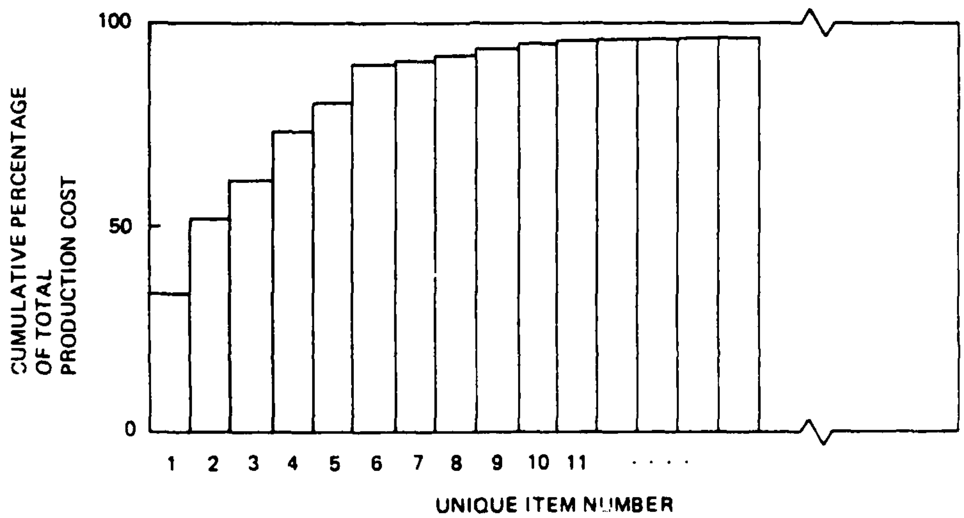


Figure D-1. Typical High-Value Item Chart

3.0 COST MODEL WITH EXAMPLES

3.1 ACQUISITION

There are few, if any, accounting and CER models for electronic hardware acquisition. It is advisable to contact the Defense Logistic Studies Information Exchange (DLSIE), U.S. Army Logistics Management Center, Fort Lee, VA 23801 (phone: AUTOVON 687-2240 or (804) 734-2240) and request a custom bibliography or catalog of acquisition and support models for electronic systems. If results are negative, the hardware acquisition costs must be developed by in-house Finance personnel on the basis of engineering and production inputs. This is often a laborious, time-consuming task, especially if several alternative hardware configurations are being considered. The RCA PRICE model is widely used by military and industrial organizations to (1) predict acquisition costs, (2) calculate cost changes because of changes in requirements (e.g., MTBF and schedule), and (3) check against manually calculated cost estimates.

Many attempts have been made to develop and validate an acquisition CER for software. The following is a simple example of a CER for software acquisition. Software acquisition (generally called software development) covers software design, coding, test debugging, and documentation. In most multiplex systems, the software attributable to multiplexing is small, even though the total software is large. If the system architecture alternatives are not widely divergent, it is possibly correct to assume the differences in cost of software acquisition among alternatives to be small. The model is

$$SW\ \$ = CL$$

where

SW $ = software development cost (dollars)
C = cost per instruction or line of code ($/instruction)
L = number of instructions or lines of code

From experience, C ranges from $30 to $200, depending on the efficiency of the software personnel and whether C represents a machine instruction or a line of code written in a high-level language. Experience on many software programs also indicates that about 40% of the cost is for design, about 20% is for actual coding, and about 40% is for test debugging and documentation (this is sometimes called the 40-20-40 rule). See tables D-1, D-2, and D-3 for a more detailed software acquisition model.

3.2 O&S

A number of accounting models are available for electronic hardware O&S costs. DLSIE is probably the best source of information. The USAF logistics support cost (LSC) model is widely used, but it requires detailed knowledge down to the line replaceable unit (LRU) level and is therefore not appropriate for use in the early phases of multiplex selection. Another approach, the Honeywell model (see table D-4) is a simplified version of LSC. Its constants may require updating because of the rapid advances in electronic technology since the model was developed. This model may be most useful for relative cost comparisons of candidate hardware configurations.

AVGSIZ	Average number of assembly-level instructions per "unit" of a package. When a unit is written in a higher level language (e.g., Fortran IV and ALGOL), the number of these higher level instructions must be adjusted upward by the user to compensate. A factor of six will be used in this event. (C)
CPMM	Cost per man-month for a programmer. (C)
CRATE	Percentage of number of instructions to be changed in package $_k$ each month. (C)
INSTMM	Number of assembly-level instructions that one man can deliver in 1 month. This number depends on the difficulty of the package $_k$ programming task.

$INSTMM_k$ if	Program is
500	Easy
250	Medium
100	Hard

	This number must be adjusted in the same manner as $AVGSIZ_k$ for higher level language. (C) (S = 500 for easy, 250 for medium, 100 for hard)
KT	Number of "packages" into which the programming system is broken down. (C)
NOUNIT	Number of "units" comprising $package_k$. (C)
OVERHD	Overhead factor to include management, secretarial help, etc. Expressed as a fraction and may be greater than 1.0. (C)
TRAIN	Cost per man to train a programmer. (C)
TRNOVR	Programmer turnover rate expressed as a fraction. $0.0 \leq TRNOVR \leq 1.0$ (C)

(C) Information provided by Contractor

Table D-1. Software Cost Variables

CA is the cost of acquisition.

$$CA = \left[(CPMM)\,(2.5) \sum_{k=1}^{KT} MANMO_k \right] (1 + OVERHD)$$

$$CSUP = \sum_{k=1}^{KT} (CRATE_k)\,(MANMO_k)$$

$$\left[(CPMM)\,(12)\,(PIUP) + \left[1 + (PUIP-1)\,(TRNOVR) \right] (TRAIN) \right]$$

In the equation for CA, the MANMO summation, which can be determined by

$$\sum_{k=1}^{KT} \frac{(NOUNIT_k)\,(AVGSIZ_k)}{(INSTMM_k)}$$

calculates the number of man-months to program package $_k$. The total cost of development (design, program, implement, and test) is considered to be 2.5 times the man-months. The cost of man-months is found by multiplying by CPMM. Finally, an overhead factor reflecting costs of secretarial help, management, etc., is multiplied.

Table D-2. Software Cost Acquisition

The cost of support (CSUP) equation consists of a summation reflecting the number of man-months to program and test package changes.

$$CSUP = \sum_{k=1}^{KT} (CRATE_k)\,(MANMO_k)$$

This is multiplied by a factor consisting of the cost per man over the life of the system plus the cost of personnel turnover.

A unit is defined as a group of computer instructions (possibly a subroutine or function). A package is defined as a group of units, and a system is a group of packages.

Table D-3. Software Support Cost

Software maintenance cost is the cost of personnel required for modification or enhancement of the multiplex software. Any special hardware (e.g., processors) required for these tasks is part of hardware acquisition. A software maintenance CER developed from experience is

$$P = L/10,000$$

where

P = number of software maintenance personnel required
L = lines of code or number of instructions

This CER assumes a central point for software maintenance (e.g., a hot bench). If maintenance is not centralized, but is conducted at B bases, then

$$P = BL/10,000$$

Again, if system architectures being compared are not widely divergent, the difference in O&S software support costs will be small. At the present time, considerable uncertainty exists for estimating software life cycle costs.

$$O\&S\ \$ = 1.8 \times 10^3 \left[\frac{NLRU}{12} + \frac{7}{NLRU}\right]\left(\frac{FHM}{MTBF}\right) N$$ base repair over 10 years

$$+ \left[PF_{RSS} - PF_{ISS}\right] C^1{}_{PA}$$ 10 years of replenishment spares

$$+ 3.5 \times 10^4 \left(\frac{FHM}{MTBF}\right)\left(\frac{N}{NLRU}\right)$$ 10 years of depot repair

$$+ 3.2 \times 10^3 + (1.5 \times 10^4)(NLRU)$$ 10 years of supply system management

NLRU = number of LRU's per system

FHM = flight-hours per month for all installed systems (hours)

MTBF = system MTBF (hours)

PF_{ISS} = production learning curve factor for number N_1 of installed + initial spare systems, where $N_1 = N\left(1 + \frac{4}{3}\frac{FHM}{MTBF} \cdot \frac{1}{NLRU}\right)$

PF_{RSS} = production learning curve factor for number N_2 of installed + initial + replenishment spare systems, where $N_2 = N_1 + 4N\left(\frac{FHM}{MTBF} \cdot \frac{1}{NLRU}\right)$

N = number of installed systems

$C^1{}_{PA}$ = cost to produce 1,000 systems

Reference: "Life Cycle Cost Comparisons of Avionic System Design Alternatives," P. S. Kilpatrick and A. L. Jones, Honeywell, Inc., p. 514-520, *NAECON '74 Record.*

Table D-4. Honeywell Avionics Simplified O&S Cost Model

4.0 UNCERTAINTY CONSIDERATIONS

Output accuracy of any cost prediction depends on accuracy of its inputs. Multiplex design decisions based on LCC comparisons can easily be misleading if the uncertainties in the LCC estimates are not taken into account (e.g., an apparent 5% difference in LCC of two candidate multiplex systems is not necessarily true if the LCC values themselves are only accurate to ± 10%). Consequently, every input to the LCC estimating process should include its numerical or percentage uncertainty. Then, the effect of these uncertainties should be considered by calculating the best case and worst case costs (LCC, acquisition, or O&S) to get the output uncertainties associated with the base LCC estimate. This procedure not only helps prevent erroneous LCC-based decisions but also calls attention to those cost elements where more accuracy is required.

Cost risk is defined as the probability that a current cost estimate will exceed a target previously set. Therefore, cost risk is a measure of the uncertainties in the input values that make up a current estimate. Cost risk is valuable to program managers since it forewarns them of the need to reallocate resources (i.e., it helps to minimize program panics and to meet contractual commitments). Cost risk is a "one-tailed" statistical test that is based on the view that a current estimate is a mean value rather than a single number. Since the estimate is a mean, it can be associated with a probability distribution. The area under the distribution curve beyond the target value (i.e., for costs exceeding the target) is then the cost risk probability. The model contains two fundamental assumptions:

a. Each cost estimate is statistically independent of the others. This assumption allows a cost risk to be calculated for an item that consists of more than one element (each separately costed).

b. Cost estimates are normally distributed. This assumption allows a cost risk probability distribution curve to be generated.

The method is quite simple. Consider an item that is built as an assembly of three elements. The item has a specified target (e.g., average production cost). There is an uncertainty in each cost factor input to an element cost, which allows each element's cost uncertainty to be calculated. The root sum square (RSS) of these uncertainties is the 1-sigma uncertainty of the item's predicted cost. Twice this value is the 2-sigma cost uncertainty. A normal distribution curve is constructed with this 2-sigma value, with the cost estimate as the mean. The target value is marked on the distribution. The area under the curve for costs exceeding the target value then represents the probability that the target value will be exceeded. Figure D-2 summarizes the results in a manner easily understood by nontechnical personnel. This technique can be applied to any item (hardware, software, or task). It is best applied during concept validation and full-scale development, when input data requirements are reasonably well defined and when their values and associated uncertainties can be justified. The technique can be used during concept definition and in the early period of validation to establish preliminary cost targets.

Uncertainties can be greatly magnified if cost models are used whose input level of detail exceeds the current state of knowledge of the multiplex system characteristics. For example, if one is at the black-box block

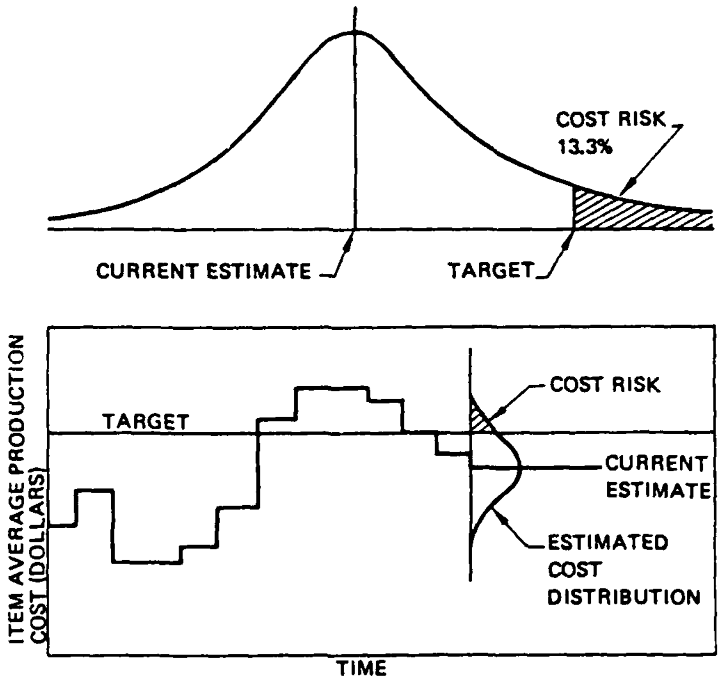


Figure D-2. Cost Risk

diagram phase of design, then it is inappropriate to use a cost model that requires component-level (e.g., types and quantities) inputs. A good general rule is to select a model whose input level is compatible with system knowledge. This immediately implies that the cost models used will change as the program progresses to ever-more-detailed design. Table D-5 summarizes the categories and applications of the models discussed in this section.

Table D-5. Summary of Cost Prediction Model Types and Applications

Model				Application															
	Type			System								Predicts costs of			Best used during				
Name	Accounting	CER	Other	Airplane	ASM	SAM	SSM	Space	Ground	Avionics	Software	Development	Production	O&S	Definition	Validation	Full-scale development	Production	O&S
Logistic support cost (AFLC)	X			X	X	X	X	X	X	X				X			X		X
PRICE (RCA)			X	X	X	X	X	X	X	X		X	X			X	X		
Honeywell		X								X				X	X	X			
Software acquisition		X									X	X	X		X	X			
Software maintenance		X									X			X	X	X			
Cost risk			X	X	X	X	X	X	X	X	X		X			X	X		

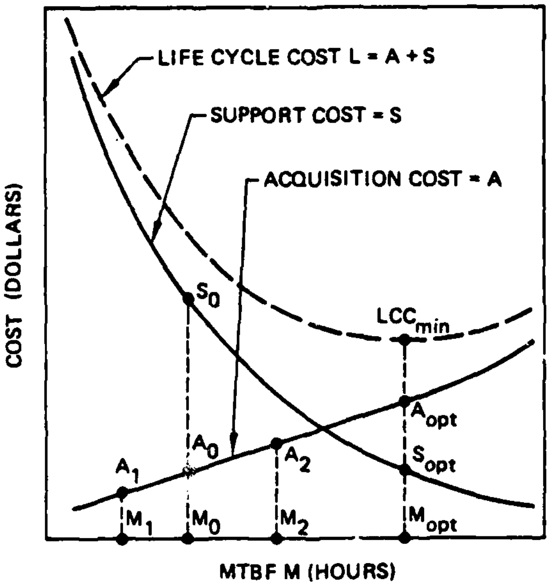
5.0 AUXILIARY RELIABILITY MODELS

5.1 RELIABILITY AND MTBF PREDICTION

Accurate reliability prediction is essential to determine mission effectiveness. Accurate MTBF prediction is also a prerequisite for accurate determination of spares requirements and prediction of unscheduled maintenance man-hours per flight-hour or maintenance man-hours per operating hour. A new and realistic prediction model for jet aircraft and their subsystems is summarized in table D-6. This model was developed from field data and has since been confirmed by additional field data. A complete description of the model is in the reference cited in table D-6. The two inputs are lambda-zero, the in-flight steady-state failure rate and T (the nominal sortie duration). Lambda-zero can be developed from failure rate handbooks. T is a design or performance characteristic of the airplane that will contain the multiplex system.

5.2 COST AND MTBF OPTIMIZATION

Cost versus MTBF will probably be one of the most important trade studies for a multiplex system. Initially, the trade will help set the MTBF goal and approximate funding requirements for the system. Subsequent trades, with alternative configurations, will help identify the preferred candidate design. After the configuration is selected, the trade will be repeated with more detailed and accurate inputs to determine if the goal will be achieved and also to reveal the possible need for reliability improvement and/or funding reallocations. Figure D-3 lists the equations with which the minimum LCC (LCC_{min}), its corresponding optimum MTBF (M_{opt}), and the relative uncertainty in LCC_{min} can be calculated. The necessary inputs are the initial estimate of support cost, S_o; three estimates of acquisition cost A_o, A_1, and A_2; and the corresponding MTBFs M_o, M_1, and M_2. The values for S_o, A_o, A_1, and A_2 are developed by Finance or from CER or forecast sensitivity models.



Given

S_0 and A_0 estimates for an arbitrary MTBF M_0

A_1 and A_2 estimates for M_1 and M_2 straddling M_0

Then

$$S(M) = \frac{k}{A_0} M_0 \qquad \text{where } k = \left(\frac{S_0}{A_0}\right) M_0$$

$$A(M) = A_0\left(\frac{1 - R_0}{1 - R}\right)^n \approx A_0\left(\frac{M}{M_0}\right)^n \qquad \text{where } n = \frac{1}{2}\left[\frac{\ln (A_1/A_0)}{\ln (M_1/M_0)} + \frac{\ln (A_2/A_0)}{\ln (M_2/M_0)}\right]$$

$$LCC_{min} = \left(\frac{K}{nM_0}\right)^{\frac{n}{n+1}} (1 + n)\, A_0$$

$$M_{opt} = \left(\frac{kM_0{}^n}{n}\right)^{\frac{1}{n+1}}$$

and the relative uncertainty in the predicted minimum LCC, LCC_{min}, is

$$\frac{\Delta\, LCC_{min}}{LCC_{min}} = \pm \sqrt{\left[\frac{n}{(n+1)^2} (\ln k - \ln n - \ln M_0)\right]^2 \left(\frac{\Delta n}{n}\right)^2 + \left(\frac{n}{n+1}\right)^2 \left[\left(\frac{\Delta k}{k}\right)^2 + \left(\frac{\Delta M_0}{M_0}\right)^2\right] + \left(\frac{\Delta A_0}{A_0}\right)^2}$$

Figure D-3. Cost Versus MTBF Optimization Equations

Item	Equation
Failure rate after t hours into sortie, $\lambda(t)$	$\lambda(t) = \frac{0.45\lambda_o T}{t + 0.016T} = \frac{k}{t + 0.016T}$
Reliability after t hours into sortie, $R(t)$	$R(t) = \left(1 + \frac{t}{0.016T}\right)^{-k}$
Expected number of failures after t hours into sortie, $E(F_t)$	$E\left(F_t\right) = k \ln\left(1 + \frac{t}{0.016T}\right)$
Mean flight time between failures for sorties of duration τ, $MTBF(\tau)$	$MTBF(\tau) = \frac{\tau}{k \ln\left(1 + \frac{\tau}{0.016T}\right)}$
Unscheduled maintenance man-hours per flight-hour for τ-hour sorties, $\frac{MMH}{FH}(\tau)$	$\frac{MMH}{FH}(\tau) = \frac{\overline{(MMH/FIX)}\, k \ln\left(1 + \frac{\tau}{0.016T}\right)}{\tau}$

λ_o = in-flight steady-state failure rate (failures/hour)

T = nominal sortie duration (hours)

t = time into sortie (hours)

$\overline{MMH/FIX}$ = average MMH/FIX for unscheduled maintenance

Table D-6. Reliability and MTBF Prediction Model for Jet Aircraft and Their Subsystems

U.S.Government Printing Office: 1981 — 757-002/455